

**Ein normiertes Penaltyfunktionsverfahren
zur Lösung des
nichtlinearen Optimierungsproblems
mit Nebenbedingungen**

**Inaugural-Dissertation
zur Erlangung der Doktorwürde
des Fachbereichs Mathematik
der Freien Universität Berlin**

vorgelegt von
H. Roland Weistroffer
aus Herne

1976



Digitized by the Internet Archive
in 2026

Ein normiertes Penaltyfunktionsverfahren zur Lösung des nichtlinearen Optimierungsproblems mit Nebenbedingungen

**Inaugural-Dissertation
zur Erlangung der Doktorwürde
des Fachbereichs Mathematik
der Freien Universität Berlin**

vorgelegt von
H. Roland Weistroffer
aus Herne

1. Referent: Prof. Dr. U. Schendel
2. Referent: Prof. Dr. H.-J. Töpfer

Tag der mündlichen Prüfung: 12. 8. 1976

Tag der Promotion:



Ich danke Herrn Professor Dr. Udo Schendel für seine Betreuung bei der Entstehung der vorliegenden Arbeit, insbesondere für die vielen wertvollen Tips und Anregungen, mit denen er mich zu jeder Zeit bereitwillig unterstützte.

Herrn Professor Dr. F.A. Lootsma, Technische Hochschule Delft, danke ich für das Interesse, welches er meiner Arbeit entgegenbrachte und für seine briefliche Unterstützung.

INHALT

	Seite
I. EINLEITUNG	7
1.1 Das nichtlineare Optimierungsproblem	7
1.2 Idee der Penaltyfunktionen	9
1.3 Inhaltsübersicht	11
II. BEKANNTE PENALTYVERFAHREN	12
2.1 Die parametrische Barriere-Funktion	12
2.2 Die parametrische Loss-Funktion	13
2.3 Die gemischte parametrische Penaltyfunktion	14
2.4 Die Barriere-Funktion mit veränderlichen Stützungen	15
2.5 Die Loss-Funktion mit veränderlichen Stützungen	17
III. EINE NORMIERTE PENALTYFUNKTION	21
3.1 Einführende Bemerkungen	21
3.2 Die normierte Loss-Funktion mit veränderlichen Stützungen	23
3.3 Anwendung des Verfahrens	31
3.4 Fehlerabschätzung und Konvergenzgeschwindigkeit	33
IV. BESCHREIBUNG DES ALGORITHMUS	40
4.1 Zusammenfassung des Gesamtalgorithmus	40
4.2 Minimierung ohne Nebenbedingungen	45
4.3 Lineare Suche	47
4.4 Numerische Berechnung der Ableitungen	50
4.5 Extrapolation gegen das Minimum des folgenden Minimierungszyklus ohne Nebenbedingungen	51
V. PRAKTISCHE ERFAHRUNGEN	54
5.1 Bemerkungen zu NEMT	54
5.2 Beschreibung des Testverfahrens	55
5.3 Ergebnisse	57
5.4 Broyden - Fletcher - Shanno versus Davidon - Fletcher - Powell	63
5.5 Einige zusätzliche Ergebnisse	65

	Seite
SCHLUSSFOLGERUNG	72
LITERATURVERZEICHNIS	73
ANHANG A: Beispiel eines nichtlinearen Optimierungs- problems mit Nebenbedingungen	77
ANHANG B: Testaufgaben	81
ANHANG C: FORTRAN-Subroutine NEMT	99
LEBENS LAUF	125

I. EINLEITUNG

1.1. Das nichtlineare Optimierungsproblem

In der vorliegenden Arbeit werden Verfahren zur Lösung nichtlinearer Optimierungsaufgaben behandelt. Das allgemeine nichtlineare Optimierungsproblem läßt sich so formulieren:

$$\begin{array}{ll} f(x) \rightarrow \text{Min} & \\ \text{mit } g_i(x) \geq 0 & ; \quad (i=1, \dots, m) \end{array} \quad \left. \vphantom{\begin{array}{l} f(x) \rightarrow \text{Min} \\ g_i(x) \geq 0 \end{array}} \right\} \quad (1.1)$$

wobei f, g_i Abbildungen von $\mathbb{R}^n$ nach $\mathbb{R}$ sind (nicht notwendig linear).

In Worten ausgedrückt besagt (1.1), daß unter den Punkten $x \in \mathbb{R}^n$, für die gilt, daß alle $g_i(x)$ ($i=1, \dots, m$) nichtnegativ sind, der Vektor gesucht wird, für den $f(x)$ minimal wird.

$f(x)$ heißt Zielfunktion, und die $g_i(x)$ ($i=1, \dots, m$) heißen Nebenbedingungen. Ist $m = 0$, so handelt es sich um ein unbedingtes Optimierungsproblem.

Die Beschränkung auf das Minimierungsproblem in (1.1) bedeutet keinen Verlust an Allgemeinheit, da sich wegen

$$\text{Max } f(x) = - \text{Min } (-f(x)) \quad (1.2)$$

jedes Maximierungsproblem in ein Minimierungsproblem überführen läßt. Gleichungen als Nebenbedingungen können wie folgt als zwei Ungleichungen aufgefaßt werden:

$$g(x) = 0 \iff \begin{cases} g(x) \geq 0 \\ -g(x) \geq 0 \end{cases} \quad (1.3)$$

Ein praxisbezogenes Beispiel eines nichtlinearen Optimierungsproblems mit Nebenbedingungen ist in Anhang A angegeben.

Def. 1.1: Die Menge

$$B := \{x / x \in \mathbb{R}^n \text{ und } g_i(x) \geq 0; i=1, \dots, m\} \quad (1.4)$$

heißt zulässiger Bereich von (1.1). Ein Element $x \in B$ heißt zulässiger Punkt oder zulässige Lösung.

Def. 1.2: Die optimale zulässige Lösung von (1.1), also der Punkt $x \in B \subset \mathbb{R}^n$, für den $f(x)$ minimal wird, heißt bedingtes Minimum von (1.1). Um Mißverständnisse zu vermeiden, wird ein Optimum im herkömmlichen Sinne, also das Optimum einer Funktion, ohne Berücksichtigung von Nebenbedingungen, in der vorliegenden Arbeit auch unbedingtes Optimum (Minimum oder Maximum) genannt.

Def. 1.3: Eine Funktion $h(z) : \mathbb{R}^n \rightarrow \mathbb{R}$ heißt konvex, wenn für alle $z^{(1)}, z^{(2)} \in \mathbb{R}^n$ und λ aus $(0,1)$ gilt:

$$h(\lambda z^{(1)} + (1-\lambda)z^{(2)}) \leq \lambda h(z^{(1)}) + (1-\lambda)h(z^{(2)}) \quad (1.5)$$

Unter der Voraussetzung, daß $h(z)$ differenzierbar ist, läßt sich die Konvexität einer Funktion (äquivalent zu Def. 1.3) wie folgt definieren:

Def. 1.3a: Eine differenzierbare Funktion $h(z) : \mathbb{R}^n \rightarrow \mathbb{R}$ heißt konvex, wenn für alle $z^{(1)}, z^{(2)} \in \mathbb{R}^n$ gilt:

$$h(z^{(1)}) - h(z^{(2)}) \geq \nabla h(z^{(1)}) (z^{(2)} - z^{(1)}) \quad (1.6)$$

$\nabla h(z^{(1)})$ bezeichnet den Gradientenvektor, das heißt der Vektor der partiellen Ableitungen von $h(z)$ an der Stelle $z^{(1)}$, also

$$\nabla h(z^{(1)}) := \left[\frac{\partial h(z^{(1)})}{\partial z_1^{(1)}}, \frac{\partial h(z^{(1)})}{\partial z_2^{(1)}}, \dots, \frac{\partial h(z^{(1)})}{\partial z_n^{(1)}} \right]^T \quad (1.7)$$

wobei $z^{(1)} = [z_1^{(1)}, z_2^{(1)}, \dots, z_n^{(1)}]^T$.

Def. 1.4: Eine Funktion $h(z): \mathbb{R}^n \rightarrow \mathbb{R}$ heißt konkav, wenn $-h(z)$ konvex ist.

Def. 1.5: Das Problem (1.1) heißt konvex, wenn die Zielfunktion $f(x)$ konvex ist und die Nebenbedingungen $g_i(x)$ ($i=1, \dots, m$) konvex sind.

Die bekannten Optimierungsverfahren können im allgemeinen nur für konvexe Optimierungsprobleme bewiesen werden. In der Praxis werden sie auch für nichtkonvexe Probleme angewandt, liefern dann in der Regel aber nur ein lokales Optimum (falls sie nicht überhaupt versagen).

1.2 Idee der Penaltyfunktionen

Grundidee der Penalty-("Straf"-)Funktionsverfahren ist, die Zielfunktion zu "bestrafen", wenn eine oder mehrere Nebenbedingungen verletzt sind. Man unterscheidet zwischen äußeren ("echten") Penaltyfunktionen und inneren Penaltyfunktionen.

Bei den äußeren Penaltyfunktionen wird der Zielfunktion außerhalb des zulässigen Bereichs ein Strafglied angehängt. Wenn dieses Strafglied ausreichend groß ist, wird das Verfahren so gezwungen, eine Lösung innerhalb des zulässigen Bereichs zu finden. Äußere Penaltyfunktionen werden auch Loss-Funktionen ("Verlust"-Funktionen) genannt.

Innere Penaltyfunktionen sind nur im Inneren des zulässigen Bereichs definiert. Es wird hierbei der Zielfunktion ein Glied angehängt, welches über alle Maße wächst, sobald es sich dem Rand des zulässigen Bereichs nähert. Innere Penaltyfunktionen heißen deshalb auch Barriere-Funktionen. Der Nachteil von Barriere-Funktionen gegenüber Loss-Funktionen ist offensichtlich: Um eine innere Penaltyfunktion zu verwenden, muß der zulässige Bereich ein nichtleeres Inneres besitzen und es muß ein innerer zulässiger Punkt von vornherein bekannt sein. Nebenbedin-

gungen wie (1.3) sind folglich nicht erlaubt.

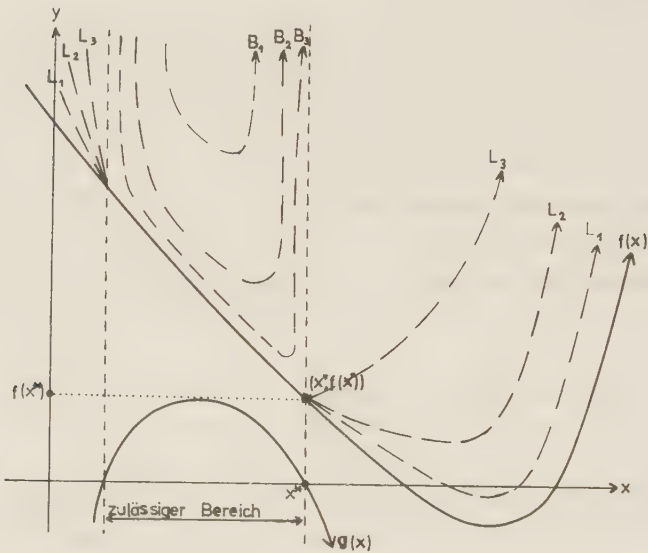


Abb. 1.1

In Abb. 1.1 ist das Prinzip der inneren und der äußeren Penaltyfunktionen für den eindimensionalen Fall veranschaulicht. $f(x)$ ist die zu minimierende Zielfunktion und $g(x)$ ist Nebenbedingung. Der zulässige Bereich ist der Abschnitt auf der Abszisse, für den $g(x)$ positiv ist. Das bedingte Minimum liegt auf dem Rand des zulässigen Bereichs und ist mit x^* bezeichnet. L_1, L_2 und L_3 sind Loss-Funktionen und B_1, B_2, B_3 sind Barriere-Funktionen. Aus Abb. 1.1 ist zu erkennen, wie die Minima einer Folge von Loss-Funktionen sich von außerhalb des zulässigen Bereichs dem bedingten Minimum von $f(x)$ nähern, bzw. wie die Minima einer Folge von Barriere-Funktionen sich dem bedingten Minimum von innerhalb des zulässigen Bereichs nähern.

Durch Anwendung von Penaltyfunktionen läßt sich also das allgemeine

nichtlineare Optimierungsproblem (1.1) auf eine Folge von unbedingten Optimierungsproblemen reduzieren. Bekannte Verfahren zur Optimierung ohne Nebenbedingungen können angewandt werden.

1.3 Inhaltsübersicht

In Abschnitt II dieser Arbeit werden einige wichtige bekannte Penalty-funktionen beschrieben, darunter auch die Loss-Funktion mit veränderlichen Stützungen ("moving truncations"). Die in Abschnitt III eingeführte normierte Penaltyfunktion ist eine Weiterentwicklung der Loss-Funktion mit veränderlichen Stützungen und unterscheidet sich von dieser nur durch bestimmte Gewichte (Normen), die der Zielfunktion sowie den Nebenbedingungen zugefügt werden. Abschnitt IV enthält eine Beschreibung des vorgeschlagenen Verfahrens und in Abschnitt V werden die praktischen Versuchsergebnisse mit der normierten Penaltyfunktion zusammengestellt und erläutert. Das zugehörige FORTRAN-Programm und die zum Testen verwendeten Aufgaben sind im Anhang dieser Arbeit enthalten.

II. BEKANNTE PENALTYVERFAHREN

Bei den folgenden Beschreibungen bekannter Penaltyfunktionsverfahren wurden die Beweise bewußt ausgelassen. Die Konvergenzaussagen und Beweise in Kapitel 3.2 lassen sich zum Teil auf die hier beschriebenen Verfahren übertragen. Ansonsten sei auf die angeführte Literatur verwiesen.

2.1 Die parametrische Barriere-Funktion

Die allgemeine Form der parametrischen inneren Penaltyfunktion kann wie folgt dargestellt werden:

$$B_R(x) := f(x) + R \cdot \sum_{i=1}^m \varphi[g_i(x)] \quad (2.1)$$

Es sind $f(x)$ und $g_i(x)$ ($i=1, \dots, m$) wie in (1.1). R ist ein positiver Parameter, und $\varphi(\eta)$ ist eine stetig differenzierbare Funktion von $\mathbf{R}^+$ nach $\mathbf{R}$ mit der Eigenschaft

$$\varphi(\eta) \rightarrow \infty \text{ für } \eta \rightarrow 0 \quad (2.2)$$

Es folgt also, daß die Funktion (2.1) im Inneren des zulässigen Bereichs definiert ist, aber auf jedem Randpunkt singulär ist.

Wenn (1.1) konvex ist, existiert (siehe [3]) für jedes $R > 0$ ein Punkt $x(R)$, der (2.1) im Inneren des zulässigen Bereichs minimiert.

Sei $\{R_k; k=1, 2, \dots\}$ eine monoton fallende Nullfolge. Dann ist ein Grenzpunkt von $\{x(R_k); k=1, 2, \dots\}$ optimale zulässige Lösung von (1.1).

Das bedingte Minimum von (1.1) läßt sich also durch sequentielles Minimieren (ohne Nebenbedingungen) der Penaltyfunktion (2.1) finden.

Zwei häufig erwähnte und auch in der Praxis angewandte Beispiele für (2.1) sind:

a) Die logarithmische Barriere-Funktion

$$BL_R(x) := f(x) - R \cdot \sum_{i=1}^m \ln[g_i(x)] \quad (2.3)$$

b) Die inverse Barriere-Funktion

$$BI_R(x) := f(x) + R \cdot \sum_{i=1}^m \frac{1}{g_i(x)} \quad (2.4)$$

Für ausführlichere Information über parametrische innere Penaltyfunktionen sei auf [3], [12], [13], [16] und [17] verwiesen.

2.2 Die parametrische Loss-Funktion

Äußere Penaltyfunktionen müssen, zum Unterschied von Barriere-Funktionen, auf dem gesamten $\mathbb{R}^n$ definiert sein. Die allgemeine parametrische Loss-Funktion läßt sich so darstellen:

$$L_R(x) := f(x) + \frac{1}{R} \cdot \sum_{i=1}^m \psi[g_i(x)] \quad (2.5)$$

$f(x), g_i(x)$ ($i = 1, \dots, m$) sind wieder wie in (1.1). R ist ein positiver Parameter, und $\psi(\eta)$ ist eine stetig differenzierbare Funktion von $\mathbb{R}$ nach $\mathbb{R}$, mit der Eigenschaft:

$$\left. \begin{array}{l} \psi(\eta) = 0 \text{ für } \eta \geq 0 \\ \text{und } \psi(\eta) > 0 \text{ für } \eta < 0 \end{array} \right\} \quad (2.6)$$

In Abb. 2.1 ist ein Beispiel für $\psi(\eta)$ dargestellt. Abb. 2.2 zeigt das Verhalten von $\psi[g(x)]$ im eindimensionalen Fall.

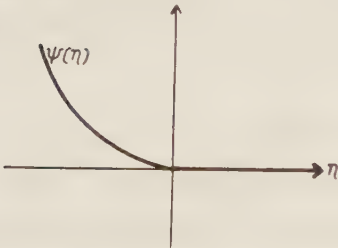


Abb. 2.1

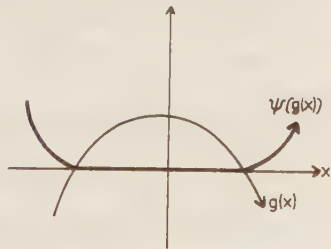


Abb. 2.2

Das Gewicht des Strafgliedes ist groß für kleine Werte von R . Wenn (1.1) konvex ist, existiert ein Punkt $x(R)$, welcher $L_R(x)$ über $\mathbb{R}^n$ minimiert.

In der Regel liegt $x(R)$ außerhalb des zulässigen Bereichs, jedoch ist der Abstand von $x(R)$ zum zulässigen Bereich desto geringer, je kleiner R gewählt wird ($R > 0$). Sei $\{R_k; k=1, 2, \dots\}$ eine monoton fallende Nullfolge, dann ist ein Grenzpunkt von $\{x(R_k); k=1, 2, \dots\}$ optimale zulässige Lösung von (1.1). Das allgemeine Optimierungsproblem (1.1) ist wieder auf eine Folge von unbedingten Optimierungsproblemen reduziert worden.

Das wichtigste Beispiel für (2.5) ist die quadratische Loss-Funktion:

$$LQ_R(x) := f(x) + \frac{1}{R} \cdot \sum_{i=1}^m \text{Min}^2 [0, g_i(x)] \quad (2.7)$$

Die parametrische äußere Penaltyfunktion ist in [3], [12], [16] und [17] beschrieben.

2.3 Die gemischte parametrische Penaltyfunktion

Wie schon gezeigt wurde, haben innere Penaltyfunktionen gegenüber äußeren Penaltyfunktionen den Nachteil, daß der zulässige Lösungsbereich ein nichtleeres Inneres haben muß. Außerdem muß von einem Punkt im Inneren des zulässigen Bereichs gestartet werden. Andererseits ist es aber auch von Vorteil, wenn, wie bei der Barriere-Funktion, der zulässige Lösungsbereich zu keinem Zeitpunkt verlassen wird. Es genügt folglich, bei Anwendung innerer Penaltyfunktionen, daß die Zielfunktion auf dem zulässigen Lösungsbereich definiert ist, während sie bei Loss-Funktionen auf dem gesamten $\mathbb{R}^n$ definiert sein muß.

Die gemischte parametrische Penaltyfunktion

$$M_{R_1/2}(x) := f(x) + R_1 \sum_{i=1}^p \varphi[g_i(x)] + \frac{1}{R_2} \sum_{i=p+1}^m \psi[g_i(x)] \quad (2.8)$$

vereint wenigstens zum Teil die Vorteile von inneren und von äußeren Penaltyfunktionen. Es sind $f(x), g_i(x)$ ($i=1, \dots, m$) wieder die Problemfunktionen von (1.1). O.B.d.A. gilt für die Nebenbedingungen am Startpunkt

$x^{(0)}$ der Berechnungen

$$\begin{aligned} g_i(x^{(0)}) &> 0 \text{ für } i=1, \dots, p \\ g_i(x^{(0)}) &\leq 0 \text{ für } i=p+1, \dots, m \end{aligned} \quad (2.9)$$

$\varphi(\eta)$ und $\psi(\eta)$ sind stetig differenzierbare Funktionen von $\mathbb{R}^+$ nach $\mathbb{R}$, bzw. von $\mathbb{R}$ nach $\mathbb{R}$, mit der Eigenschaft (2.2), bzw. (2.6). R_1 und R_2 sind positive Parameter ($R_1=R_2$ möglich). Für $p=0$ reduziert sich (2.8) zu (2.5); wir erhalten die parametrische Loss-Funktion. Für $p=m$ erhalten wir die parametrische Barriere-Funktion (2.1).

Für das konvexe Optimierungsproblem gibt es einen Punkt $x(R_1, R_2)$, der $M_{R_{1/2}}(x)$ minimiert. Seien $\{R_{1,k}; k=1, 2, \dots\}$ und $\{R_{2,k}; k=1, 2, \dots\}$ monoton fallende Nullfolgen, so konvergiert $\{x(R_{1,k}, R_{2,k}); k=1, 2, \dots\}$ gegen die optimale zulässige Lösung.

Die gemischte parametrische Penaltyfunktion vereint den Vorteil des beliebigen Startpunktes mit der Eigenschaft, daß die Nebenbedingungen, die am Startpunkt erfüllt sind, während des gesamten Verlaufs der Minimierung erfüllt bleiben.

LOOTSMA ([16], [20]) benutzt für die von ihm entwickelten Optimierungsalgorithmen die folgende gemischte parametrische Penaltyfunktion:

$$MLQ_R(x) := f(x) - R \cdot \sum_{i=1}^p \ln[g_i(x)] + \frac{1}{R} \cdot \sum_{i=p+1}^m \text{Min}^2[0, g_i(x)] \quad (2.10)$$

Die Penaltyfunktion (2.8) ist unabhängig von LOOTSMA [14] und von FIACCO und McCORMICK [3] entwickelt worden.

2.4 Die Barriere-Funktion mit veränderlichen Stützungen

Im vorhergehenden wurde gezeigt, wie mit Hilfe von Penaltyfunktionen das allgemeine Optimierungsproblem durch eine Folge von unbedingten Minimierungen gelöst werden kann. Diese sequentiellen Minimierungen sollen im folgenden als Zyklen bezeichnet werden.

Die Optimierungsverfahren mit veränderlichen Stützungen (moving trunca-

tions) beruhen auf der Idee, mit Hilfe der Zielfunktion, als eine weitere künstliche Nebenbedingung, den zulässigen Bereich nach jedem Zyklus zu verkleinern.

Def. 2.1: $f(x)$ sei die Zielfunktion von (1.1) und B der zulässige Bereich (siehe (1.4)). Der durch hinzufügen der zusätzlichen Nebenbedingung

$$t - f(x) \geq 0 \quad (2.11)$$

bestimmte "gestutzte" zulässige Bereich sei mit $T(t)$ bezeichnet:

$$T(t) := \{x / f(x) \leq t; x \in B\} \quad (2.12)$$

t heißt die zugehörige "Stutzebene" (truncation level).

Um $T(t) \neq \emptyset$ zu gewährleisten, darf t nicht kleiner als der Wert der minimalen zulässigen Lösung von (1.1) sein.

Die allgemeine Barriere-Funktion mit veränderlichen Stutzungen läßt sich nun wie folgt darstellen:

$$TB_t(x) := \varphi[t - f(x)] + \sum_{i=1}^m \varphi[g_i(x)] \quad (2.13)$$

$\varphi(\eta)$ ist eine konvexe Funktion von $\mathbb{R}^+$ nach $\mathbb{R}$, nicht steigend im Intervall $(0, \infty)$, mit der Eigenschaft (2.2).

Bezeichnung: Sei A eine Menge. Dann ist A° der offene Kern oder das Innere von A .

Wenn (1.1) konvex ist, existiert ein Punkt $x(t) \in T^\circ(t)$, der (2.13) minimiert. Dieses Minimum soll im folgenden "Zentrum" von $T(t)$ genannt werden (in Anlehnung an die "Zentrenmethode" von HUARD, [8]) und wird mit $c_t := x(t)$ bezeichnet. (Die Minimierung von (2.13) kann als Minimierung der parametrischen Barriere-Funktion (2.1) aufgefaßt werden, wobei $\varphi[t - f(x)]$ die Zielfunktion ist und $R=1$ angenommen wird; die Existenz eines Zentrums c_t folgt also aus der Existenz eines minimierenden Punktes $x(R)$ für (2.1)). Es folgt aus der Konstruktion von (2.13), daß

$$f(c_t) < t \quad (2.14)$$

gilt, da wegen (2.2) $TB_t(x)$ sonst unendlich groß würde. Sei x^* die optimale zulässige Lösung von (1.1). Dann läßt sich eine monoton fallende Folge $\{t_k; k=1,2,\dots\}$ konstruieren, deren Grenzwert gleich $f(x^*)$ ist, und es gilt (siehe [16]):

$$\lim_{k \rightarrow \infty} c_{t_k} = \lim_{t \rightarrow f(x^*)} c_t = x^* \quad (2.15)$$

Eine solche Folge $\{t_k\}$ erhält man zum Beispiel durch die Iterationsformel

$$t_{k+1} = t_k - \varphi [t_k - f(c_{t_k})] ; (k=1,2,\dots) \quad (2.16)$$

wobei $\varphi \in (0,1]$. $\varphi = 1$ empfiehlt sich jedoch nicht, da dann das Zentrum c_{t_k} nicht im Inneren, sondern auf dem Rand von $T(t_{k+1})$ liegt und folglich nicht als Startpunkt für den nächsten Zyklus verwendet werden kann.

Ein interessantes Beispiel für (2.13) ist

$$TBL_t(x) := -\ln [t-f(x)] - \sum_{i=1}^m \ln [g_i(x)] \quad (2.17)$$

Diese Funktion über $T^0(t)$ zu minimieren ist äquivalent dazu, die Funktion

$$d_t(x) := [t-f(x)] \cdot \prod_{i=1}^m g_i(x) \quad (2.18)$$

über $T(t)$ zu maximieren. (2.18) ist ein Beispiel der "Distanzfunktion" der Zentrenmethode von HUARD ([8], [12]). Die Distanzfunktionen haben die Eigenschaft, daß sie im Inneren des gestutzten zulässigen Bereichs ($T^0(t)$) positiv sind und auf den Rand von $T(t)$ verschwinden.

Literatur zur Barriere-Funktion mit veränderlichen Stützungen: [3], [15], [16], [17].

2.5 Die Loss-Funktion mit veränderlichen Stützungen

Die Loss-Funktion mit veränderlichen Stützungen unterscheidet sich von der Barriere-Funktion mit veränderlichen Stützungen dadurch, daß Ver-

fahren, welche auf erstere beruhen, sich der optimalen zulässigen Lösung von (1.1) von außerhalb des gestutzten zulässigen Bereichs $T(t)$ nähern. Die allgemeine äußere Penaltyfunktion mit veränderlichen Stutzungen wird folgendermaßen dargestellt:

$$TL_t(x) := \psi[t - f(x)] + \sum_{i=1}^m \psi[g_i(x)] \quad (2.19)$$

$f(x)$, $g_i(x)$ ($i=1, \dots, m$) sind wie immer Zielfunktion und Nebenbedingungen von (1.1). $\psi(\eta)$ ist eine konvexe Funktion von $\mathbb{R}$ nach $\mathbb{R}$, nichtsteigend, mit der Eigenschaft (2.6). Es ist leicht zu sehen, daß

$$\left. \begin{array}{l} TL_t(x) = 0 \quad \text{für } x \in T(t) \\ \text{und } TL_t(x) > 0 \quad \text{für } x \notin T(t) \end{array} \right\} \quad (2.20)$$

Aus (2.12) folgt, daß $T(t)$ genau dann nicht leer ist, wenn der zulässige Bereich B nicht leer ist und t nicht kleiner ist als der Wert der minimalen zulässigen Lösung x^* von (1.1). Es gilt also, daß $TL_t(x) > 0$ für alle $x \in \mathbb{R}^n$, falls $t < f(x^*)$.

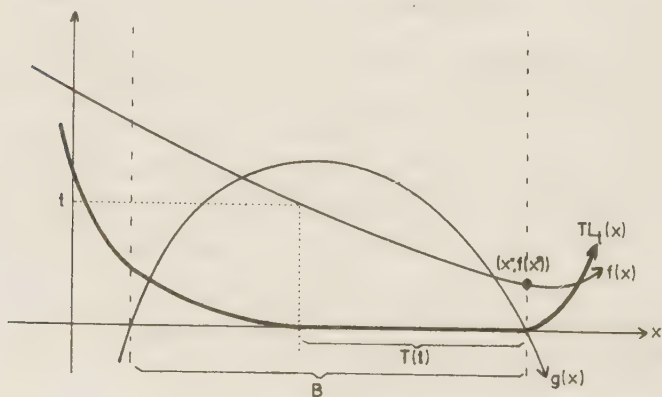


Abb. 2.3

In Analogie zur Barriere-Funktion mit veränderlichen Stutzungen soll das Minimum c_t von $TL_t(x)$ Zentrum von $T(t)$ genannt werden. Für $t > f(x^*)$

ist jeder Punkt aus $T(t)$ Zentrum, da dann $TL_t(x) = 0$ für alle $x \in T(t)$ gilt. (Siehe Abb. 2.3). Das heißt, um ein eindeutiges Zentrum von $T(t)$ zu erhalten, muß $t \leq f(x^*)$ sein. Ist $t = f(x^*)$, dann ist c_t optimale zulässige Lösung von (1.1). Da $f(x^*)$ im allgemeinen aber nicht bekannt ist, muß von $t < f(x^*)$ ausgegangen werden. Für diesen Fall läßt sich folgende wichtige Ungleichung zeigen:

$$t < f(c_t) \leq f(x^*) \quad (2.21)$$

Das "Zentrum" liegt also außerhalb von $T(t)$ (es gilt sogar $T(t) = \emptyset$) und im allgemeinen ist c_t auch nicht im zulässigen Bereich B (außer wenn $f(x)$ ein unbedingtes Minimum in B besitzt). Wenn $f(c_t) = f(x^*)$, dann ist c_t optimale zulässige Lösung von (1.1). Anderenfalls läßt sich eine monoton nichtfallende Folge $\{t_k; k=1, 2, \dots\}$ mit

$$t_k < t_{k+1} \leq f(x^*) \quad (2.22)$$

finden, so daß

$$\lim_{k \rightarrow \infty} t_k = f(x^*) \quad (2.23)$$

ist. Weiter läßt sich zeigen, daß dann

$$\lim_{k \rightarrow \infty} c_{t_k} = \lim_{t \rightarrow f(x^*)} c_t = x^* \quad (2.24)$$

minimale zulässige Lösung von (1.1) ist. (Siehe Kapitel 3.2; die Beweise der Sätze 3.1 bis 3.4 lassen sich direkt auf die Penaltyfunktion (2.19) übertragen).

Ein Beispiel für (2.19) ist die Funktion

$$TLQ_t(x) = \text{Min}^2[0, t-f(x)] + \sum_{i=1}^m \text{Min}^2[0, g_i(x)] \quad (2.25)$$

Ungleichung (2.21) suggeriert folgende Beziehung zur Generierung der Folge $\{t_k\}$:

$$t_{k+1} = f(c_{t_k}) \quad (2.26)$$

$$\text{bzw. } t_{k+1} = t_k + \varphi[f(c_{t_k}) - t_k]; \quad (k=1, 2, \dots) \quad (2.27)$$

wobei $\rho \in (0,1]$ (vergleiche (2.16)).

Praktische Erfahrungen mit dieser Folge von Stutzebenen haben sich jedoch als enttäuschend erwiesen; das Verfahren konvergiert zu langsam (LOOTSMA [15]).

STAHA [26] hat die folgende Iterationsformel zur Bestimmung der Folge $\{t_k\}$ entwickelt:

$$t_{k+1} := t_k - \frac{\hat{TLQ}(t_k)}{\hat{TLQ}'(t_k)} \quad ; \quad (k=1,2,\dots) \quad (2.28)$$

$\hat{TLQ}'(t_k) = 2 [t_k - f(c_{t_k})]$ ist die Ableitung (siehe [26]) von $\hat{TLQ}(t) := TLQ_t(c_t)$ nach t an der Stelle t_k ausgewertet. Das heißt, die Funktion (2.25) wird für $x=c_t$ als Funktion der Stutzebene t aufgefaßt und nach dieser abgeleitet. Es sei bemerkt, daß (2.28) dem Newtonschen Näherungsverfahren entspricht. STAHA benutzt zur Bestimmung der Folge von Stutzebenen jeweils das Maximum aus (2.26) und (2.28) und erreicht eine wesentliche Verbesserung der Konvergenz gegenüber der Anwendung von nur (2.26).

Zwei weitere Formeln zur iterativen Bestimmung der Stutzebenen sind ([11], [19]):

$$t_{k+1} := t_k - \frac{TLQ_{t_k}(c_{t_k})}{t_k - f(c_{t_k})} \quad ; \quad (k=1,2,\dots) \quad (2.29)$$

$$t_{k+1} := t_k + \sqrt{TLQ_{t_k}(c_{t_k})} \quad ; \quad (k=1,2,\dots) \quad (2.30)$$

Die durch (2.29) bestimmten Stutzebenen heißen "Tangentsparameter" und die durch (2.30) bestimmten Stutzebenen heißen "Morrison-Parameter".

Die Loss-Funktion mit veränderlichen Stutzungen ist außer in der schon erwähnten Literatur in [16] und [17] behandelt.

III. EINE NORMIERTE PENALTYFUNKTION

3.1 Einführende Bemerkungen

Während man die jeweiligen Vorteile der inneren und der äußeren parametrischen Penaltyfunktion in der gemischten parametrischen Penaltyfunktion (wenigstens teilweise) vereinigt hat, scheint ein analoges Vorgehen bei den Penaltyfunktionen mit veränderlichen Stützungen nicht möglich zu sein. Nach Kenntnis des Autors ist es bisher noch nicht gelungen, ein Verfahren zu konstruieren, welches auf einer Kombination der Barriere-Funktion mit veränderlichen Stützungen und der Loss-Funktion mit veränderlichen Stützungen beruht. Der Grund hierfür mag darin liegen, daß man hierzu sowohl eine Folge von Stutzebenen benötigen würde, die sich dem Wert der optimalen zulässigen Lösung von (1.1) von oben nähert, als auch eine Folge von Stutzebenen, die sich dem Wert der optimalen zulässigen Lösung von unten nähert.

Um nur noch eine einzige Folge $\{t_k; k=1,2,\dots\}$ von Stutzebenen zu benötigen, könnte man die Loss-Funktion mit veränderlichen Stützungen mit der parametrischen Barriere-Funktion kombinieren, bzw. die Barriere-Funktion mit veränderlichen Stützungen mit der parametrischen Loss-Funktion kombinieren:

$$P_1(x) = \psi[t-f(x)] + R \cdot \sum_{i=1}^p \varphi[g_i(x)] + \sum_{i=p+1}^m \psi[g_i(x)] \quad (3.1)$$

$$P_2(x) = \varphi[t-f(x)] + \sum_{i=1}^p \varphi[g_i(x)] + R^{-1} \cdot \sum_{i=p+1}^m \psi[g_i(x)] \quad (3.2)$$

$\varphi(\eta)$ und $\psi(\eta)$ sind wie in Abschnitt II definiert (mit der Eigenschaft (2.6) bzw. (2.2)). $f(x)$ und $g_i(x)$ ($i=1,\dots,m$) sind die Problemfunktionen von (1.1), und R ist ein kleiner positiver Parameter. O.B.d.A. gelte

$$\left. \begin{array}{l} g_i(x^{(0)}) > 0 \quad \text{für } i = 1, \dots, p \\ g_i(x^{(0)}) \leq 0 \quad \text{für } i = p+1, \dots, m \end{array} \right\} \quad (3.3)$$

am Startpunkt $x^{(0)}$ der Berechnungen.

Die Schwierigkeit bei der Anwendung von (3.1) oder (3.2) ist aber wieder, eine Folge von Stutzebenen so zu konstruieren, daß (bei Anwendung von (3.1))

$$t_k < t_{k+1} \leq f(x^*) \quad (3.4)$$

erfüllt ist, bzw. (bei Anwendung von (3.2)) die Ungleichung

$$f(x^*) \leq t_{k+1} < t_k \quad (3.5)$$

gilt. Die wichtigen Beziehungen (2.21) bzw. (2.14) sind bei diesen gemischten Penaltyfunktionen nicht notwendig erfüllt.

Der Vorteil einer gemischten Penaltyfunktion gegenüber einer reinen Loss-Funktion liegt darin, daß einmal erfüllte Nebenbedingungen erfüllt bleiben. Bei einer reinen Loss-Funktion könnte man dies aber auch erreichen, indem man den schon erfüllten Nebenbedingungen ein schweres Gewicht hinzufügt:

$$P_3(x) = \psi[t-f(x)] + M \cdot \sum_{i=1}^p \psi[g_i(x)] + \sum_{i=p+1}^m \psi[g_i(x)] \quad (3.6)$$

M ist eine große positive Zahl. (3.6) unterscheidet sich von der gewöhnlichen Loss-Funktion mit veränderlichen Stutzungen (2.19) nur durch diesen Gewichtungsfaktor M.

Alle praktischen Versuche mit (3.6), wobei

$$\psi(\eta) := \text{Min}^2 [0, \eta] \quad (3.7)$$

gewählt wurde, verliefen jedoch enttäuschend. Der Gewichtungsfaktor M verzerrt die Funktion (3.6) und erschwert dadurch die Minimierung ohne Nebenbedingungen erheblich. Die Loss-Funktion mit veränderlichen Stutzungen (2.19) ist der Penaltyfunktion (3.6) im allgemeinen überlegen.

Anstelle eines pauschalen Gewichts M benutzt die normierte Loss-Funktion mit veränderlichen Stutzungen individuelle Gewichte für die einzelnen Nebenbedingungen sowie für die Zielfunktion. Diese individuellen

Gewichte werden durch die Kehrwerte der Problemfunktionen nach Vollendung jedes unbedingten Minimierungszyklus bestimmt. Die so gewichtete Penaltyfunktion erweist sich der Funktion (2.19) weit überlegen, auch wenn die ursprüngliche Absicht, die Erhaltung der einmal erfüllten Nebenbedingungen, nicht erreicht ist. (Siehe Abschnitt V).

3.2 Die normierte Loss-Funktion mit veränderlichen Stützungen

Da die optimale zulässige Lösung des nichtlinearen Optimierungsproblems (1.1) durch sukzessives Minimieren der Penaltyfunktion für eine Folge von Stutzebenen erreicht wird, soll im weiteren gleich eine Folge von normierten Loss-Funktionen mit veränderlichen Stützungen betrachtet werden:

$$F_k(x) := \psi[N_{0,k} \cdot (t_k - f(x))] + \sum_{i=1}^m \psi[N_{i,k} \cdot g_i(x)] \quad ; (k=1,2,\dots) \quad (3.8)$$

$f(x)$ ist die Zielfunktion und $g_i(x)$ ($i=1,\dots,m$) sind die Nebenbedingungen von (1.1). Es gelte folgendes:

a) ψ ist eine konvexe Funktion von $\mathbb{R}$ nach $\mathbb{R}$ mit der Eigenschaft

$$\left. \begin{array}{l} \psi(\eta) = 0 \quad \text{für} \quad \eta \geq 0 \\ \text{und} \quad \psi(\eta) = 0 \quad \text{für} \quad \eta < 0 \end{array} \right\} \quad (3.9)$$

b) Die $N_{i,k}$ ($i=0,\dots,m$; $k=1,2,3,\dots$) sind positive Gewichte, bestimmt durch

$$N_{0,k} := \left\{ \begin{array}{ll} m_A \cdot |f(c_{k-1})|^{-1} & \text{für} \quad 1 \leq |f(c_{k-1})| < L \\ m_A \cdot L^{-1} & \text{für} \quad |f(c_{k-1})| \geq L \\ N_{0,k-1} & \text{für} \quad |f(c_{k-1})| < 1 \end{array} \right\} \quad (3.10)$$

L ist ein großer positiver Wert, der verhindern soll, daß $N_{0,k}$ zu klein wird. m_A gibt die Anzahl der in c_{k-1} aktiven Nebenbedingungen an.

(Falls $m_A = 0$ wird $m_A := 1$ gesetzt.) c_{k-1} minimiert die normierte Loss-Funktion $F_{k-1}(x)$ und wird Zentrum genannt.

$$N_{i,k} := \begin{cases} |g_i(c_{k-1})|^{-1} & \text{für } -1 \leq g(c_{k-1}) < -\epsilon \\ \epsilon^{-1} & \text{für } g(c_{k-1}) \geq -\epsilon \\ N_{i,k-1} & \text{für } g(c_{k-1}) < -1 \end{cases} \quad (i=1, \dots, m) \quad (3.11)$$

ϵ ist ein kleiner positiver Wert, der Division durch sehr kleine Zahlen verhindern soll.

Es sei $N_{i,0} := 1$ ($i=0,1,\dots,m$). Bezüglich des Wertes von c_0 sei auf Kapitel 3.3 verwiesen.

Die durch (3.10) und (3.11) definierten Gewichte $N_{i,k}$ ($i=0,1,\dots,m$; $k=1,2,\dots$) werden fortan die zugehörigen Normen von (3.8) genannt.

c) $\{t_k; k=1,2,\dots\}$ ist eine reelle Folge, bestimmt durch

$$\left. \begin{array}{l} t_1 \leq f(x^*) \\ \text{und } t_{k+1} := f(c_k) \end{array} \right\} \quad (3.12)$$

x^* ist die optimale zulässige Lösung von (1.1). Die t_k ($k=1,2,\dots$) heißen Stutzebenen für (3.8).

Aus a) folgt, daß $\psi(\eta)$ in $\mathbb{R}$ monoton nichtsteigend und auf dem Intervall $(-\infty, 0)$ monoton fallend ist.

Das oben beschriebene Verfahren konvergiert gegen die optimale zulässige Lösung von (1.1) (Sätze 3.1 bis 3.4) unter den folgenden Bedingungen:

Bedingung (1): Das Optimierungsproblem (1.1) ist konvex.

(Siehe Def. 1.5).

Bedingung (2): Die Zielfunktion sowie die Nebenbedingungen sind stetig-differenzierbar.

Bedingung (3): Der zulässige Bereich ist kompakt und nicht leer.

Satz 3.1:

Für jedes $t_k \in \mathbb{R}$ existiert unter den Bedingungen (1) und (3) ein Punkt $c_k \in \mathbb{R}^n$ derart, daß

$$F_k(c_k) = \min_{x \in \mathbb{R}^n} F_k(x) \quad (3.13)$$

Beweis: Es ist

$$F_k(x) = \psi[N_{0,k}(t_k - f(x))] + \sum_{i=1}^m \psi[N_{i,k}g_i(x)] \quad (3.14)$$

wobei $g_i(x)$ ($i=1, \dots, m$) konkav und $f(x)$ konvex ist.

Aus $f(x)$ konvex folgt $-f(x)$ ist konkav und also ist $(t_k - f(x))$ konkav.

Sei

$$g_0(x) := t_k - f(x) \quad (3.15)$$

dann können wir (3.14) jetzt als

$$F_k(x) = \sum_{i=0}^m \psi[N_{i,k}g_i(x)] \quad (3.16)$$

schreiben, wobei alle $g_i(x)$ ($i=0, 1, \dots, m$) konkav sind.

Da der zulässige Bereich B nach Voraussetzung nicht leer ist, folgt, daß keine der Funktionen $g_i(x)$ ($i=1, \dots, m$) nur negative Werte annimmt.

Wegen der Kompaktheit und folglich Beschränktheit von B gilt aber, daß mindestens ein $g_i(x)$, $i \in \{1, 2, \dots, m\}$ negativ wird, wenn der Betrag von x in irgendeiner Komponente sehr groß wird. Da $g_i(x)$ konkav ist, folgt, daß $g_i(x)$ unbeschränkt klein wird, wenn der Betrag von x in einer Komponente unbeschränkt wächst:

$$\|x\| \rightarrow \infty \Rightarrow N_{i,k}g_i(x) \rightarrow -\infty \quad (3.17)$$

für mindestens ein $i \in \{1, \dots, m\}$.

Die Funktion $\psi(\eta)$ ist nach Voraussetzung konvex und aus (3.9) folgt, daß $\psi(\eta)$ auf den Intervall $(-\infty, 0)$ monoton fallend ist. Ähnlich wie (3.17) gilt deshalb:

$$\eta \rightarrow -\infty \Rightarrow \psi(\eta) \rightarrow \infty \quad (3.18)$$

Sei $i \in \{0, 1, \dots, m\}$, dann gilt wegen der Konkavität (für beliebige $x_1, x_2 \in \mathbb{R}^n, \lambda \in (0, 1)$):

$$g_i(\lambda x_1 + (1-\lambda)x_2) \geq \lambda g_i(x_1) + (1-\lambda)g_i(x_2) \quad (3.19)$$

Da $\psi(\eta)$ auf $\mathbb{R}$ monoton nichtsteigend ist, folgt

$$\psi[g_i(\lambda x_1 + (1-\lambda)x_2)] \leq \psi[\lambda g_i(x_1) + (1-\lambda)g_i(x_2)] \quad (3.20)$$

Aus der Konvexität von $\psi(\eta)$ folgt andererseits:

$$\psi[\lambda g_i(x_1) + (1-\lambda)g_i(x_2)] \leq \lambda \cdot \psi[g_i(x_1)] + (1-\lambda) \cdot \psi[g_i(x_2)] \quad (3.21)$$

Aus (3.20) und (3.21) zusammen erhält man

$$\psi[g_i(\lambda x_1 + (1-\lambda)x_2)] \leq \lambda \cdot \psi[g_i(x_1)] + (1-\lambda) \cdot \psi[g_i(x_2)] \quad (3.22)$$

also ist $\psi[g_i(x)]$ konvex.

Die Konstanten $N_{i,k}$ ($i=0, 1, \dots, m$) wurden außeracht gelassen, sie verändern jedoch das Ergebnis nicht. Da $i \in \{0, 1, \dots, m\}$ beliebig gewählt war, ist $\psi[N_{i,k} \cdot g_i(x)]$ konvex für alle $i=0, 1, \dots, m$ und folglich gilt:

$$F_k(x) = \sum_{i=0}^m \psi[N_{i,k} \cdot g_i(x)] \text{ ist konvex.}$$

Aus (3.18) zusammen mit (3.17) folgt nun:

$$\|x\| \rightarrow \infty \Rightarrow F_k(x) \rightarrow \infty \quad (3.23)$$

Sei a beliebig aus $\mathbb{R}^n$. Wegen (3.23) existiert ein $M \in \mathbb{R}$ derart, daß

$$\|x\| > M \Rightarrow F_k(x) > F_k(a) \quad (3.24)$$

Sei

$$S := \{x / x \in \mathbb{R}^n \text{ und } \|x\| \leq M\} \quad (3.25)$$

S ist eine abgeschlossene Kugel in $\mathbb{R}^n$ und folglich kompakt. Aus der Kompaktheit folgt: Es existiert $c_k \in S \subset \mathbb{R}^n$ derart, daß für alle $x \in S$

$$F_k(c_k) \leq F_k(x) \quad (3.26)$$

Aus (3.24) folgt, daß $\|a\| \leq M$ und deshalb $a \in S$. Es ist also $F_k(c_k) \leq F_k(a)$ und folglich

$$F_k(c_k) \leq F_k(x) \text{ für alle } x \in \mathbb{R}^n.$$

q.e.d.

Satz 3.2:

Es sei x^* optimale zulässige Lösung von (1.1) und

$$v^* := f(x^*) \quad (3.27)$$

Dann gilt für beliebiges $t_k \in \mathbb{R}$ unter der Bedingung (2):

$$t_k < v^* \Rightarrow t_k < f(c_k) \leq v^* \quad (3.28)$$

Beweis: Zunächst werden die folgenden Aussagen benötigt. Da der Gradient von $F_k(x)$ im Zentrum c_k verschwindet (notwendige Bedingung für ein Optimum), gilt:

$$\psi'[t_k - f(c_k)] \cdot N_{0,k}[-\nabla f(c_k)] + \sum_{i=1}^m \psi'[N_{i,k} \cdot g_i(c_k)] \cdot N_{i,k} \cdot \nabla g_i(c_k) = 0 \quad (3.29)$$

Ferner ist wegen $F_k(c_k) \leq F_k(x^*)$ und $\psi[N_{i,k} \cdot g_i(x)] = 0$ für x zulässig:

$$\psi[N_{0,k} \cdot (t_k - f(c_k))] + \sum_{i=1}^m \psi[N_{i,k} \cdot g_i(c_k)] \leq \psi[N_{0,k} \cdot (t_k - f(x^*))] \quad (3.30)$$

oder anders geschrieben:

$$\sum_{i=1}^m \psi[N_{i,k} \cdot g_i(c_k)] \leq \psi[N_{0,k} \cdot (t_k - v^*)] - \psi[N_{0,k} \cdot (t_k - f(c_k))] \quad (3.31)$$

Der Beweis von Satz (3.2) wird nun durch gegenteilige Annahme und Widerspruch erbracht:

a) Annahme $t_k < v^*$ und $f(c_k) \leq t_k$. Dann folgt aus (3.29)

$$\sum_{i=1}^m \psi'[N_{i,k} \cdot g_i(c_k)] \cdot N_{i,k} \cdot \nabla g_i(c_k) = 0 \quad (3.32)$$

da $\psi[t_k - f(x)] = 0$ für $f(x) \leq t_k$.

Der Punkt $c_k \in \mathbb{R}^n$ minimiert also $\sum_{i=1}^m \psi[N_{i,k} \cdot g_i(x)]$. Daraus folgt aber,

daß c_k im zulässigen Bereich liegt und folglich $f(c_k) \geq v^*$ ist. Dies ist ein Widerspruch zur Annahme und es gilt also: $t_k < v^* \Rightarrow t_k < f(c_k)$.

b) Annahme $t_k < v^*$ und $f(c_k) > v^*$. Dann ist

$$[t_k - f(c_k)] < [t_k - v^*] < 0 \quad (3.33)$$

Da $\psi(\eta)$ monoton fallend ist, folgt aus (3.33):

$$\psi[N_{0,k}(t_k - f(c_k))] > \psi[N_{0,k}(t_k - v^*)] > 0 \quad (3.34)$$

Wegen (3.31) und (3.34) gilt nun:

$$\sum_{i=1}^m \psi[N_{i,k} \cdot g_i(c_k)] < 0 \quad (3.35)$$

Dies widerspricht der Definition von $\psi(\eta)$ (siehe (3.9)), also gilt notwendig:

$$t_k < v^* \Rightarrow f(c_k) \leq v^*.$$

q.e.d.

Satz 3.3:

Besitzt die Zielfunktion $f(x)$ ein unbedingtes zulässiges Minimum, und ist $t_k < v^*$, dann ist, unter den Voraussetzungen von Satz 3.2, das Zentrum c_k optimale zulässige Lösung des Problems (1.1).

Beweis: Die Annahme, daß $f(x)$ ein unbeschränktes Minimum im zulässigen Bereich besitzt, bedeutet, daß für alle $x \in \mathbb{R}^n$ gilt: $v^* \leq f(x)$. Aus Satz 3.2 ist aber bekannt, daß $f(c_k) \leq v^*$ ist, also muß $f(c_k) = v^*$ gelten.
q.e.d.

Satz 3.4:

Die Bedingungen (1), (2) und (3) seien erfüllt. Besitzt $f(x)$ kein unbedingtes zulässiges Minimum, so daß die optimale zulässige Lösung von (1.1) auf dem Rand des zulässigen Bereichs liegt, und ist die Folge $\{t_k; k=1,2,\dots\}$ wie (3.12) bestimmt, dann konvergiert $\{t_k\}$ gegen v^* und jeder Grenzpunkt der Folge $\{c_k; k=1,2,\dots\}$ ist optimale zulässige Lösung des Problems (1.1).

Beweis: Zur Erinnerung seien die Beziehungen aus (3.12) noch einmal wiedergegeben:

$$\left. \begin{aligned} t_1 &\leq v^* \\ t_{k+1} &= f(c_k) ; (k=1,2,\dots) \end{aligned} \right\} \quad (3.36)$$

Angenommen, $f(c_k) = v^*$, so folgt aus (3.31) und der Definition von $\psi(\eta)$ (siehe (3.9)), daß c_k zulässig ist. Gleichung (3.29) wird dann zu

$$\psi' [t_k - v^*] \cdot N_{0,k} \cdot [-\nabla f(c_k)] = 0 \quad (3.37)$$

so daß gilt:

$$\nabla f(c_k) = 0 \quad (3.38)$$

Das bedeutet, daß $f(x)$ an der Stelle c_k ein unbedingtes Minimum besitzt, im Widerspruch zur Voraussetzung des Satzes.

Es ist also $f(c_k) < v^*$, und aus (3.28) folgt, daß die Folge $\{t_k, k=1,2,\dots\}$ monoton steigend und durch v^* beschränkt ist. Es sei

$$t^* := \lim_{k \rightarrow \infty} t_k \quad (3.39)$$

und es gelte $t^* < v^*$. Wegen (3.28) ist dann $t^* < f(c_{t^*})$, (c_{t^*} sei das zu t^* gehörende Zentrum) und es folgt aus der Konstruktion der Folge $\{t_k\}$, daß t^* nicht ihr Grenzwert sein kann. Es muß also gelten:

$$\lim_{k \rightarrow \infty} t_k = v^* \quad (3.40)$$

Aus (3.40) zusammen mit (3.31) ergibt sich:

$$\lim_{k \rightarrow \infty} \sum_{i=1}^m \psi [N_{i,k} \cdot g_i(c_k)] = 0 \quad (3.41)$$

Zur Weiterführung des Beweises wird der folgende Hilfssatz benötigt:

Hilfssatz:

Die Funktionen $g_i(x)$ ($i=1,\dots,m$) seien konkav, und der von ihnen bestimmte zulässige Bereich B sei kompakt und nicht leer. Dann ist die

Menge

$$B_{\delta} := \{x / x \in \mathbb{R}^n \text{ und } g_i(x) \geq -\delta_i ; i=1, \dots, m\} \quad (3.42)$$

kompakt für jede "Verwischung" der Nebenbedingungen $\delta := (\delta_1, \dots, \delta_m)^T$.

Beweis: Es genügt zu zeigen, daß B_{δ} beschränkt ist für $\delta = \bar{\delta} = (\bar{\delta}_1, 0, \dots, 0)^T$, $\bar{\delta}_1 > 0$.

Annahme B_{δ} ist nicht beschränkt. Sei $x^{(1)} \in B$ ein zulässiger Punkt. Es kann nun eine gerade Linie gefunden werden, die in $x^{(1)}$ beginnt und den Rand von B schneidet, aber nicht den Rand von $B_{\bar{\delta}}$. Sei $x^{(2)}$ ein Punkt auf dieser Linie, so daß

$$\left. \begin{aligned} g_1(x^{(2)}) &= -\alpha < 0 \\ g_i(x^{(2)}) &\geq 0 ; i=2, 3, \dots, m \end{aligned} \right\} \quad (3.43)$$

Weiter sei w ein Punkt auf dieser Linie derart, daß

$$x^{(2)} = \lambda w + (1-\lambda)x^{(1)} ; \lambda \in (0, 1) \quad (3.44)$$

Wegen der Konkavität von $g_1(x)$ gilt:

$$g_1(x^{(2)}) \geq \lambda g_1(w) + (1-\lambda)g_1(x^{(1)}) \geq \lambda g_1(w) \quad (3.45)$$

Einsetzen von (3.43) in (3.45) ergibt:

$$g_1(w) \leq \frac{-\alpha}{\lambda} \quad (3.46)$$

Es ist $w \in B_{\bar{\delta}}$ für jedes $\lambda \in (0, 1)$. Indem λ klein genug gewählt wird, ergibt sich aus (3.46) der Widerspruch

$$g_1(w) < -\bar{\delta}_1 \quad (3.47)$$

$B_{\bar{\delta}}$ ist also beschränkt.

Sei $\delta > 0$. Der Hilfssatz hat gezeigt, daß

$$B_{\delta} := \{x / x \in \mathbb{R}^n \text{ und } g_i(x) \geq -\delta ; i=1, \dots, m\} \quad (3.48)$$

kompakt ist. Wegen (3.41) gilt dann: Es gibt ein $K \in \mathbb{N}$ derart, daß

$$c_k \in B_{\delta} \text{ für alle } k \geq K \quad (3.49)$$

Daraus folgt, daß die Folge $\{c_k; k=1,2,\dots\}$ von Zentren einen Grenzwert c^* besitzt. $c^* = x^*$ ist optimale zulässige Lösung des Problems (1.1).

q.e.d.

3.3 Anwendung des Verfahrens

In Kapitel 3.2 wurde ein Optimierungsverfahren zur Lösung des konvexen Problems (1.1) beschrieben, welches auf sequentieller Minimierung einer normierten Loss-Funktion beruht, sowie die Konvergenz dieses Verfahrens bewiesen. Zur praktischen Anwendung des Algorithmus werden benötigt:

- a) Eine konvexe Funktion $\psi(\eta)$, welche die Bedingung (3.9) erfüllt.
- b) Eine erste Stutzebene $t_1 \leq f(x^*)$.

Eine Möglichkeit, die Ausgangsstutzebene $t_1 \leq v^*$ zu erhalten, ist wie folgt:

Da die parametrische äußere Penaltyfunktion (siehe Kapitel 2.2)

$$F_0(x) := f(x) + \frac{1}{R} \cdot \sum_{i=1}^m \psi[g_i(x)] \quad (3.50)$$

keine vorher bekannte Stutzebene benötigt, und für den Parameter R ein kleiner positiver Wert willkürlich gewählt werden kann, wird statt der in Kapitel 3.2 beschriebenen normierten Penaltyfunktion (3.8) im ersten Zyklus (Null-Zyklus) die Penaltyfunktion (3.50) minimiert. Sei

$$F_0(c_0) := \min_{x \in \mathbb{R}^n} F_0(x) \quad (3.51)$$

dann gilt:

$$F_0(c_0) \leq F_0(x^*) \quad (3.52)$$

Da die optimale zulässige Lösung von (1.1), x^* , zulässig ist, gilt $g_i(x^*) \geq 0$ ($i=1,\dots,m$) und folglich

$$\sum_{i=1}^m \psi[g_i(x^*)] = 0 \quad (3.53)$$

Es ist also $F_0(x^*) = f(x^*)$ und (3.52) wird zu

$$F_0(c_0) \leq f(x^*) = v^* \quad (3.54)$$

Der Wert

$$t_1 := F_0(c_0) \quad (3.55)$$

ist also eine geeignete Ausgangsstutzebene. In allen weiteren Zyklen (t-Zyklen) kann nun das Verfahren, so wie es in Kapitel 3.2 beschrieben ist, angewandt werden.

Für die Funktion $\psi(\eta)$ seien folgende Beispiele gegeben:

1) Die quadratische Funktion

$$\psi(\eta) := [\text{Min}(0, \eta)]^2 \quad (3.56)$$

2) Der hyperbolische Cosinus

$$\psi(\eta) := 2 \cosh [\text{Min}(0, \eta)] - 2 \quad (3.57)$$

$$\text{bzw. } \psi(\eta) := \exp [\text{Min}(0, \eta)] + \exp [-\text{Min}(0, \eta)] - 2 \quad (3.58)$$

$$3) \quad \psi(\eta) := |\text{Min}(0, \eta)|^p, p \in (1, 2] \quad (3.59)$$

Beispiel 1) ist ein Spezialfall von Beispiel 3) und ist die in der Praxis ([16], [20], [26]) am häufigsten verwendete Funktion. (Siehe auch (2.7), (2.10), (2.25)). Auch die praktischen Rechenvergleiche des Autors dieser Arbeit mit den Funktionen (3.56), (3.58) und (3.59), letztere für verschiedene Werte p, verliefen durchweg zugunsten von (3.56). Funktion (3.59) läßt sich noch abwandeln, indem für p in jedem Zyklus ein unterschiedlicher Wert eingesetzt wird. Aber auch diesbezügliche Versuche zeigten keinen Vorteil. Die Minimierung ohne Nebenbedingungen in jedem Zyklus ist bei Verwendung von (3.58) oder (3.59) wesentlich teurer als bei Verwendung von (3.56). Dieses Resultat überrascht nicht, wenn man die Ableitungen der drei Beispielfunktionen berechnet: Die Ableitungen von (3.58) und von (3.59) sind komplizierter als die Ableitung von (3.56). Das für die unbedingte Minimierung benutzte Quasi-Newton-Verfahren benötigt diese Ableitungen zur Bestimmung der Such-

richtung in jeder Iteration (siehe Kapitel 4.1)

3.4 Fehlerabschätzung und Konvergenzgeschwindigkeit

In diesem Kapitel werden einige Überlegungen zur Fehlerabschätzung angestellt, mit deren Hilfe sich dann auch Aussagen über die Konvergenzgeschwindigkeit machen lassen.

Um den Fehler zwischen dem, nach einer bestimmten Anzahl von Zyklen, errechneten Ergebnis und dem Wert des tatsächlichen Optimums abschätzen zu können, ist es nötig, etwas über das Verhalten der Folge von Stutzebenen $\{t_k; k=1,2,\dots\}$ zu wissen. Unter der Voraussetzung, daß $\{t_k\}$ durch (3.12) bestimmt ist, wird in Kapitel 3.2 gezeigt, daß, falls $f(x)$ kein unbedingtes zulässiges Minimum besitzt (die folgende Fehlerbetrachtung wäre dann überflüssig), die Folge der Stutzebenen monoton steigend ist und gegen $t^* = v^* = f(x^*)$ konvergiert. Daraus läßt sich schließen, daß es ein $N \in \mathbb{N}$ gibt, derart, daß für alle $k \in \mathbb{N}$ mit $k > N$ gilt:

$$\frac{t_{k+1} - t_k}{t_{N+1} - t_N} < 1 \quad (3.60)$$

Diese Eigenschaft reicht noch nicht aus, um eine brauchbare Aussage über die Größenordnung des Fehlers nach Beendigung des N -ten Zyklus zu machen. Erfüllt $\{t_k; k=1,2,\dots\}$ dagegen die stärkere Bedingung

$$\frac{t_{k+1} - t_k}{t_k - t_{k-1}} \leq K < 1 \quad (3.61)$$

für alle $k \geq N$, (K ist eine Konstante), so läßt sich der Fehler zwischen t_N und t^* abschätzen.

Satz 3.5:

$\{t_k; k=1,2,\dots\}$ sei die durch (3.12) bestimmte Folge von Stutzebenen, und es existiere ein $N \in \mathbb{N}$ derart, daß (3.61) für alle $k \geq N$ erfüllt ist.

Dann gilt:

$$t^* - t_N \leq \sum_{j=0}^{\infty} K^j (t_{N+1} - t_N) = \frac{1}{1-K} (t_{N+1} - t_N) \quad (3.62)$$

Beweis: Da $\{t_k; k=1, 2, \dots\}$ monoton steigend ist und gegen t^* konvergiert, gilt:

$$t^* - t_N = \sum_{k=N}^{\infty} (t_{k+1} - t_k) \quad (3.63)$$

Es ist also zu zeigen, daß

$$\sum_{k=N}^{\infty} (t_{k+1} - t_k) \leq \sum_{j=0}^{\infty} K^j (t_{N+1} - t_N) \quad (3.64)$$

erfüllt ist. Dies trifft offensichtlich zu, wenn für alle $j = 0, 1, 2, \dots$ folgende Ungleichung erfüllt ist:

$$t_{N+1+j} - t_{N+j} \leq K^j (t_{N+1} - t_N) \quad (3.65)$$

Für $j = 0$ gilt (3.65) trivialerweise und für $j = 1$ folgt die Ungleichung unmittelbar aus (3.61). Die allgemeine Gültigkeit von (3.65) läßt sich durch Induktion zeigen: Sei (3.65) für $j = m$ erfüllt, dann gilt

$$K \cdot (t_{N+1+m} - t_{N+m}) \leq K \cdot K^m \cdot (t_{N+1} - t_N) \quad (3.65)$$

Wegen (3.61) gilt außerdem

$$\frac{t_{N+2+m} - t_{N+1+m}}{t_{N+1+m} - t_{N+m}} \leq K \quad (3.66)$$

Einsetzen von (3.66) in (3.65) ergibt

$$t_{N+2+m} - t_{N+1+m} \leq K^{m+1} \cdot (t_{N+1} - t_N) \quad (3.67)$$

und damit ist (3.65) für $j = m+1$ gezeigt.

q.e.d.

Um ein K für die Abschätzung (3.62) zu erhalten, betrachte man die Folge

$$\left\{ \frac{t_{k+2} - t_{k+1}}{t_{k+1} - t_k}; k = 1, 2, \dots \right\} \quad (3.68)$$

In der Praxis kann davon ausgegangen werden, daß

$$\frac{t_3 - t_2}{t_2 - t_1} < 1 \quad (3.69)$$

und (3.68) zumindest für kleine k monoton nicht steigend ist.

Nach Beendung des N -ten Zyklus (und Berechnung von t_{N+1}) läßt sich zur Fehlerabschätzung der Wert

$$K = K_{N-1} := \frac{t_{N+1} - t_N}{t_N - t_{N-1}} \quad (3.70)$$

wählen.

Korollar 1 zu Satz 3.5:

Die Folge (3.68) sei monoton nicht steigend und die Ungleichung (3.69) sei erfüllt. Dann gilt:

$$t^* - t_{N-1} \leq \left[1 / \left(1 - \frac{t_{N+1} - t_N}{t_N - t_{N-1}} \right) \right] \cdot (t_N - t_{N-1}) \quad (3.71)$$

oder äquivalent dazu:

$$t^* - t_{N+1} \leq \left[1 / \left(1 - \frac{t_{N+1} - t_N}{t_N - t_{N-1}} \right) \right] \cdot (t_N - t_{N-1}) - (t_{N+1} - t_{N-1}) \quad (3.72)$$

Korollar 2 zu Satz 3.5:

Es sei

$$\frac{t_{k+2} - t_{k+1}}{t_{k+1} - t_k} = K < 1 \quad (3.73)$$

konstant für alle $k = 1, 2, \dots$. Dann gilt

$$t^* - t_N = \frac{1}{1-K} (t_{N+1} - t_N) \quad (3.74)$$

Der folgende Satz gibt eine hinreichende Bedingung für die Monotonie der Folge (3.68):

Satz 3.6:

$\{t_k ; k=1,2,\dots\}$ sei die durch (3.12) bestimmte Folge von Stutzebenen.
Die Funktion $\Phi(t) : \mathbb{R} \rightarrow \mathbb{R}$ sei definiert durch

$$\Phi(t) := \lambda \cdot t_{k+1} + (1-\lambda) \cdot t_k \quad \text{für} \quad t = \lambda \cdot t_k + (1-\lambda) \cdot t_{k-1} \quad (3.75)$$

$$(k=2,3,\dots ; \lambda \in [0,1])$$

Insbesondere ist dann

$$\Phi(t_k) = t_{k+1} \quad (3.76)$$

Sei $\Phi(t)$ konkav, dann gilt:

$$\left\{ \frac{t_{k+2} - t_{k+1}}{t_{k+1} - t_k} ; k=1,2,\dots \right\} \quad (3.77)$$

ist monoton nicht steigend.

Beweis: Aus $\Phi(t)$ konkav folgt für $\lambda \in [0,1]$:

$$\Phi(\lambda \cdot t_{k_1} + (1-\lambda) \cdot t_{k_2}) \geq \lambda \cdot \Phi(t_{k_1}) + (1-\lambda) \cdot \Phi(t_{k_2}) \quad (3.78)$$

Sei $k \in \{1,2,\dots\}$ beliebig und

$$\lambda = \frac{t_{k+1} - t_k}{t_{k+2} - t_k} \quad (3.79)$$

Dann gilt

$$\begin{aligned} \lambda \cdot t_{k+2} + (1-\lambda) \cdot t_k &= \frac{t_{k+1} - t_k}{t_{k+2} - t_k} \cdot t_{k+2} + \left(1 - \frac{t_{k+1} - t_k}{t_{k+2} - t_k}\right) \cdot t_k \\ &= \frac{t_{k+1} \cdot t_{k+2} - t_{k+1} \cdot t_k}{t_{k+2} - t_k} \\ &= t_{k+1} \end{aligned} \quad (3.80)$$

Es folgt also:

$$\Phi(t_{k+1}) = \Phi[\lambda \cdot t_{k+2} + (1-\lambda) \cdot t_k] \geq \lambda \cdot \Phi(t_{k+2}) + (1-\lambda) \cdot \Phi(t_k)$$

$$\begin{aligned}
 \Rightarrow \phi(t_{k+1}) &\geq \frac{t_{k+1} - t_k}{t_{k+2} - t_k} \cdot \phi(t_{k+2}) + \left(1 - \frac{t_{k+1} - t_k}{t_{k+2} - t_k}\right) \cdot \phi(t_k) \\
 \Rightarrow \phi(t_{k+1}) - \phi(t_k) &\geq \frac{t_{k+1} - t_k}{t_{k+2} - t_k} \cdot [\phi(t_{k+2}) - \phi(t_k)] \\
 \Rightarrow \frac{\phi(t_{k+1}) - \phi(t_k)}{t_{k+1} - t_k} &\geq \frac{\phi(t_{k+2}) - \phi(t_k)}{t_{k+2} - t_k} \\
 \Rightarrow \frac{t_{k+2} - t_{k+1}}{t_{k+1} - t_k} &\geq \frac{t_{k+3} - t_{k+1}}{t_{k+2} - t_k} \quad (3.81)
 \end{aligned}$$

Ungleichung (3.81) läßt sich weiter umformen zu:

$$\begin{aligned}
 \frac{(t_{k+3} - t_{k+2}) + (t_{k+2} - t_{k+1})}{(t_{k+2} - t_{k+1}) + (t_{k+1} - t_k)} &\leq \frac{t_{k+2} - t_{k+1}}{t_{k+1} - t_k} \\
 \Rightarrow (t_{k+3} - t_{k+2}) \cdot (t_{k+1} - t_k) &\leq (t_{k+2} - t_{k+1})^2 \\
 \Rightarrow \frac{t_{k+3} - t_{k+2}}{t_{k+2} - t_{k+1}} &\leq \frac{t_{k+2} - t_{k+1}}{t_{k+1} - t_k} \quad (3.82)
 \end{aligned}$$

q.e.d.

Das durch (3.70) bestimmte K läßt sich auch zur Abschätzung der Konvergenzgeschwindigkeit verwenden. Hierzu sei der Begriff "Konvergenzwert" definiert.

Def. 3.1: Sei $\{t_k; k=1,2,\dots\}$ die Folge der Stutzebenen, dann heißt

$$C_k := \frac{t_{k+1} - t_k}{t^* - t_k}; \quad (k=1,2,\dots) \quad (3.83)$$

Konvergenzwert im k -ten t -Zyklus (der parametrische "Null-Zyklus" (siehe Kapitel 3.3) wurde nicht mitgezählt).

Es ist klar, daß $C_k \in (0,1)$ für alle $k = 1,2,\dots$. Je näher C_k an 1, um so schneller konvergiert das Verfahren. Da t^* im allgemeinen nicht

bekannt ist, läßt sich C_k wie folgt abschätzen. Aus (3.62):

$$t'' - t_k \leq \frac{1}{1-K} \cdot (t_{k+1} - t_k) ; (k=1,2,\dots) \quad (3.84)$$

folgt:

$$C_k \geq 1 - K ; (k=1,2,\dots) \quad (3.85)$$

Unter den Bedingungen von Korollar 1 zu Satz 3.5 gilt also:

$$C_k \geq 1 - \frac{t_{k+2} - t_{k+1}}{t_{k+1} - t_k} ; (k=1,2,\dots) \quad (3.86)$$

Zur Illustration der Ergebnisse aus diesem Kapitel wurden die Beziehungen (3.72) und (3.86) auf die Testaufgabe 3 (siehe Anhang B) angewandt. Bezüglich der Durchführung der Rechnungen sei auf Abschnitt V verwiesen. Die normierte Loss-Funktion mit veränderlichen Stützungen durchlief 3 t-Zyklen bei der Bearbeitung von Aufgabe 3. Es ergab:

$$t_1 = 1.32451 ; t_2 = 1.39280 ; t_3 = 1.39347$$

Aus (3.72) erhält man:

$$t'' - t_3 \leq 0.000007$$

d.h., t_3 und $f(x'')$ unterscheiden sich maximal durch 7×10^{-6} . Für den Konvergenzwert ergibt sich aus (3.86):

$$C_1 \geq 0.99$$

Die Loss-Funktion mit veränderlichen Stützungen ohne Normierungen durchlief 12 t-Zyklen bei Bearbeitung der gleichen Aufgabe. Es seien hier nur die ersten und die letzten drei Stutzebenen angegeben:

$$t_1 = 1.32451 ; t_2 = 1.35852 ; t_3 = 1.37587$$

$$t_{10} = 1.39333 ; t_{11} = 1.39340 ; t_{12} = 1.39343$$

Daraus errechnet sich:

$$t^* - t_3 \leq 0.018069$$

$$t^* - t_{12} \leq 0.000023$$

und $C_1 \geq 0.49$

$$C_{10} \geq 0.57$$

Zu diesen Ergebnissen muß bemerkt werden, daß das für diese Berechnungen benutzte Programm (siehe Abschnitt IV) die Folge der Stutzebenen $\{t_k ; k=1,2,\dots\}$ nicht durch die Beziehung

$$t_{k+1} = f(c_k) \tag{3.87}$$

sondern jeweils durch das Maximum von den durch (3.87) und den durch (2.28) errechneten Werten bestimmt. In der Praxis bestimmte die normierte Loss-Funktion bei der Bearbeitung von Aufgabe 3 die Stutzebenen jedoch ausschließlich durch Beziehung (3.87).

IV. BESCHREIBUNG DES ALGORITHMUS

Ein Algorithmus, welcher auf dem in Kapitel 3.3 und 3.4 beschriebenen Verfahren beruht, wurde als FORTRAN - Subroutine NEMT (Normed Exterior Moving Truncation) an einem TR 440 Rechner getestet. Eine Liste des Programms ist in Anhang C dieser Arbeit enthalten.

Im folgenden wird zuerst ein Überblick über den Gesamtalgorithmus gegeben, und danach werden einige benötigte Subalgorithmen beschrieben.

4.1 Zusammenfassung des Gesamtalgorithmus

Abweichend von Kapitel 3.3 wird die Folge der Stutzebenen $\{t_k ; k=1,2,\dots\}$ wie folgt bestimmt:

$$t_{k+1} := \text{Max} (t_{k+1}^{(1)}, t_{k+1}^{(2)}) \quad (4.1)$$

wobei

$$t_{k+1}^{(1)} := f(c_k) \quad (4.2)$$

$$\text{und} \quad t_{k+1}^{(2)} := t_k - \frac{F_k(c_k)}{F'_k(c_k)} \quad (4.3)$$

Es ist hierbei c_k das Zentrum im k -ten t -Zyklus und $F'_k(c_k)$ ist die Ableitung der Penaltyfunktion nach der Stutzebene t (siehe (2.28)).

STAHA [26] beweist, daß für die quadratische Loss-Funktion mit veränderlichen Stutzungen (2.25) gilt:

$$t_k < f(x^*) \Rightarrow t_k < t_{k+1}^{(2)} \leq f(x^*) \quad (4.4)$$

Dieser Beweis läßt sich direkt auf die normierte quadratische Loss-Funktion

$$F_k(x) := \text{Min}^2 [0, N_{0,k} \cdot (t_k - f(x))] + \sum_{i=1}^m \text{Min}^2 [0, N_{i,k} \cdot g_i(x)] \quad (4.5)$$

übertragen, und Satz 3.4 behält seine Gültigkeit auch für die durch

(4.1) bestimmte Folge $\{t_k ; k=1,2,\dots\}$. Der Beweis von (4.4) wird an

dieser Stelle ausgelassen, da die Wahl von Beziehung (4.1) zur Bestimmung der Stutzebenen sich auf die Effektivität der normierten Loss-Funktion nur geringfügig auswirkt. In den praktischen Versuchen ergab sich, daß die Stutzebenen fast ausschließlich durch (4.2) bestimmt werden und nur in Ausnahmefällen durch (4.3). Der Grund, weshalb NEMT trotzdem die Beziehung (4.1) verwendet, ist folgender: Die Subroutine NEMT ist so konzipiert, daß der Benutzer wählen kann, ob die normierte Loss-Funktion (4.5) für das Verfahren benutzt wird, oder die quadratische Loss-Funktion (2.25). In letzterem Fall werden die Normen $N_{i,k}$ ($i=0,1,\dots,m$; $k=1,2,\dots$) alle gleich 1 gesetzt. Bei Verwendung der quadratischen Loss-Funktion ohne Normierungen wird der Konvergenzwert jedoch wesentlich verbessert, wenn die Stutzebenen durch (4.1) bestimmt werden (anstelle von: $t_{k+1} := f(c_k)$).

Für die normierte quadratische Loss-Funktion (4.5) ergibt sich:

$$F_k'(c_k) = 2 \cdot N_{0,k}^2 \cdot (t_k - f(c_k)) \quad \text{für} \quad t_k < f(c_k) \quad (4.6)$$

In den bisherigen theoretischen Betrachtungen wurden Nebenbedingungen in Form von Gleichungen als zwei Ungleichungen aufgefaßt (siehe (1.3)). Im folgenden werden diese jedoch getrennt behandelt. Es gilt o.B.d.A:

$$\left. \begin{array}{ll} g_i(x) \geq 0 & \text{für} \quad i = 1, \dots, p \\ \text{und} \quad g_i(x) = 0 & \text{für} \quad i = p+1, \dots, m \end{array} \right\} \quad (4.7)$$

Nach diesen einleitenden Bemerkungen können nun die einzelnen Schritte des in NEMT verwendeten Algorithmus wie folgt zusammengefaßt werden:

1.) Minimiere die parametrische äußere Penaltyfunktion

$$F_0(x) := f(x) + \frac{1}{R} \cdot \sum_{i=1}^p \text{Min}^2[0, g_i(x)] + \frac{1}{R} \cdot \sum_{i=p+1}^m g_i^2(x) \quad (4.8)$$

Der Parameter R ist vom Benutzer zu bestimmen. (Siehe auch Kapitel 2.2). In den meisten Fällen hat sich der Wert $R = 0.05$ bewährt. Der bei

der Minimierung erhaltene Punkt sei mit c_0 bezeichnet.

2.) Bestimme die Stutzebene t_1 durch

$$t_1 := F_0(c_0) \quad (4.9)$$

setze $k := 1$ und $N_{i,0} := 1$ ($i=0,1,\dots,m$).

3.) Bestimme die Norm $N_{0,k}$ der Zielfunktion durch

$$\left. \begin{aligned} N_{0,k} &:= [\text{Max}(m_A, 1)] / |f(c_{k-1})| \quad \text{für } 1 \leq |f(c_{k-1})| < 10\,000 \\ N_{0,k} &:= [\text{Max}(m_A, 1)] / 10\,000 \quad \text{für } |f(c_{k-1})| \geq 10\,000 \\ N_{0,k} &:= N_{0,k-1} \quad \text{für } |f(c_{k-1})| < 1 \end{aligned} \right\} \quad (4.10)$$

$m_A \leq m$ ist die Anzahl der in c_{k-1} aktiven Nebenbedingungen.

4.) Bestimme die Normen $N_{i,k}$ ($i=1,\dots,p$) der Ungleichungsnebenbedingungen durch

$$\left. \begin{aligned} N_{i,k} &:= |g_i(c_{k-1})|^{-1} \quad \text{für } -1 \leq g_i(c_{k-1}) < -\epsilon \\ N_{i,k} &:= \epsilon^{-1} \quad \text{für } g_i(c_{k-1}) \geq -\epsilon \\ N_{i,k} &:= N_{i,k-1} \quad \text{für } g_i(c_{k-1}) < -1 \end{aligned} \right\} \quad (4.11)$$

ϵ ist ein vom Benutzer zu bestimmender kleiner positiver Wert (empfohlen wird: $\epsilon := 0.01$).

5.) Bestimme die Normen $N_{i,k}$ ($i=p+1,\dots,m$) der Gleichungsnebenbedingungen durch

$$\left. \begin{aligned} N_{i,k} &:= |g_i(c_{k-1})|^{-1} \quad \text{für } \epsilon < |g_i(c_{k-1})| \leq 1 \\ N_{i,k} &:= \epsilon^{-1} \quad \text{für } |g_i(c_{k-1})| \leq \epsilon \\ N_{i,k} &:= N_{i,k-1} \quad \text{für } |g_i(c_{k-1})| > 1 \end{aligned} \right\} \quad (4.12)$$

ϵ wie unter Schritt 4.).

6.) Minimiere die normierte äußere Penaltyfunktion mit veränderlichen Stützungen

$$F_k(x) := \text{Min}^2 [0, N_{0,k} (t_k - f(x))] + \sum_{i=1}^p \text{Min}^2 [0, N_{i,k} g_i(x)] + \sum_{i=p+1}^m [N_{i,k} g_i(x)]^2 \quad (4.13)$$

Der bei der Minimierung erhaltene Punkt sei mit c_k bezeichnet.

7.) Bestimme die Stutzebene t_{k+1} durch

$$t_{k+1} := \text{Max} \left[t_k + \frac{F_k(c_k)}{2 \cdot N_{0,k} (f(c_k) - t_k)}, f(c_k) \right] \quad (4.14)$$

8.) Extrapoliere gegen

$$\tilde{c}_{k+1} \approx c_{k+1} \quad (4.15)$$

9.) Prüfe auf Konvergenz: Falls 8.) erfolgreich war, prüfe ob

$$|\tilde{x}_j^{(k+1)} - \bar{x}_j^{(k)}| \leq \epsilon_R |\tilde{x}_j^{(k+1)}| + \epsilon_A ; (j=1, \dots, n) \quad (4.16)$$

Falls 8.) nicht erfolgreich war, prüfe ob

$$|\bar{x}_j^{(k)} - \bar{x}_j^{(k-1)}| \leq \epsilon_R |\bar{x}_j^{(k)}| + \epsilon_A ; (j=1, \dots, n) \quad (4.17)$$

Dabei ist $c_k^T =: [\bar{x}_1^{(k)}, \dots, \bar{x}_n^{(k)}]$, $c_{k-1}^T =: [\bar{x}_1^{(k-1)}, \dots, \bar{x}_n^{(k-1)}]$ und

$\tilde{c}_{k+1}^T =: [\tilde{x}_1^{(k+1)}, \dots, \tilde{x}_n^{(k+1)}]$. ϵ_R und ϵ_A sind vom Benutzer zu bestimmende kleine positive Werte. (Empfohlen: $\epsilon_A := \epsilon_R := 5 \times 10^{-5}$).

10.) Falls 9.) erfüllt ist, ist das Verfahren beendet. Anderenfalls setze

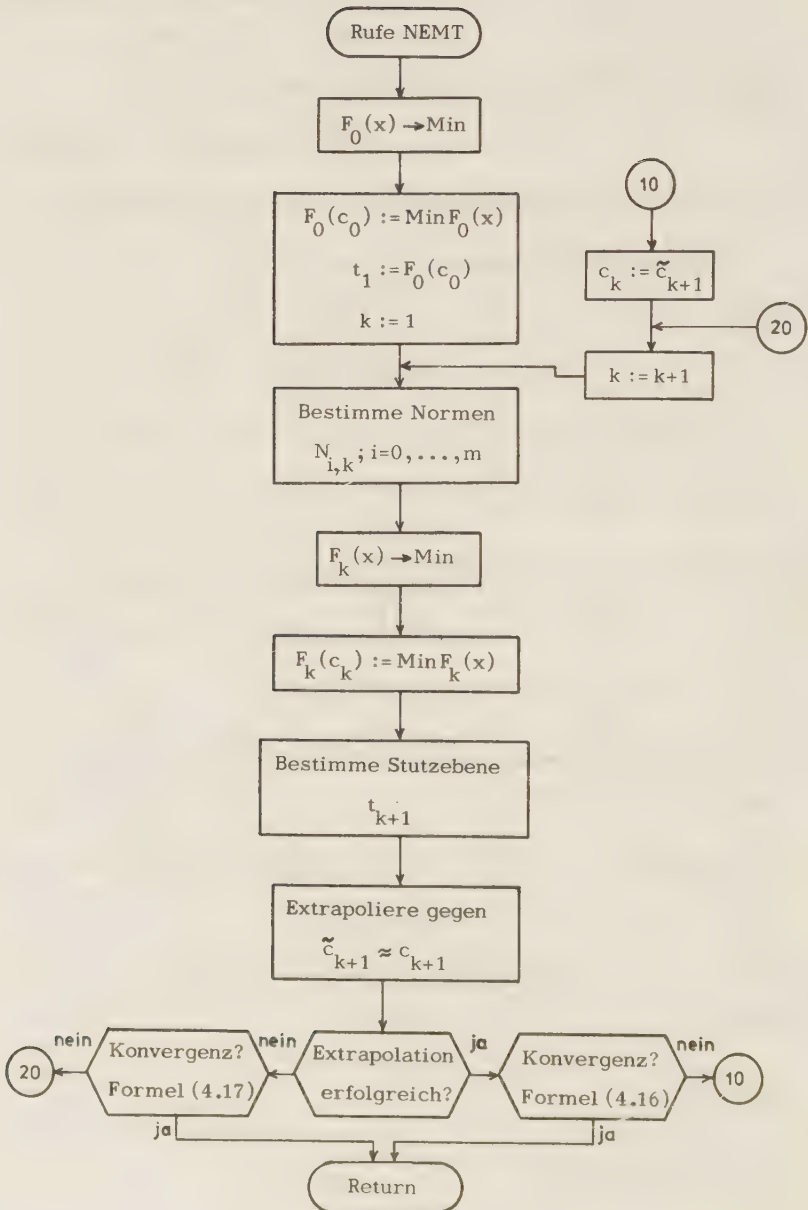
$$c_k := \tilde{c}_{k+1}, \text{ falls 8.) erfolgreich war.}$$

Setze

$$k := k+1$$

und gehe zu Schritt 3.) .

Ablaufplan für NEMT



4.2 Minimierung ohne Nebenbedingungen

Für Schritt 1.) bzw. Schritt 6.) des in Kapitel 4.1 beschriebenen Algorithmus wird ein Verfahren benötigt, welches die Penaltyfunktion minimiert. Ein sehr erfolgreiches Verfahren zur Minimierung ohne Nebenbedingungen ist der Davidon-Fletcher-Powell- (DFP-) Algorithmus, eine Quasi-Newton- oder Variable-Metric-Methode. Der DFP-Algorithmus bestimmt eine Suchrichtung $s^{(k)}$ im k -ten Iterationsschritt durch

$$s^{(k)} := -H^{(k)} \cdot g^{(k)} \quad (4.18)$$

Dabei ist $g^{(k)}$ der Gradient der zu minimierenden Funktion, also in diesem Fall der Penaltyfunktion, und $H^{(k)}$ ist die sogenannte Richtungsmatrix, eine Approximation der inversen Hesseschen Matrix. Als $H^{(0)}$ wird die Einheitsmatrix genommen, und $H^{(k+1)}$ wird wie folgt iterativ bestimmt:

$$H^{(k+1)} := H^{(k)} + \frac{\Delta x \cdot \Delta x^T}{\Delta x^T \cdot \Delta g} - \frac{H^{(k)} \cdot \Delta g \cdot \Delta g^T \cdot H^{(k)}}{\Delta g^T \cdot H^{(k)} \cdot \Delta g} \quad (4.19)$$

für $k=0,1,2,\dots$, wobei $\Delta x := x^{(k+1)} - x^{(k)}$ und $\Delta g := g^{(k+1)} - g^{(k)}$.

Die Schrittweite in jeder Iteration wird durch eindimensionale Minimierung in Suchrichtung bestimmt. Es ist

$$x^{(k+1)} := x^{(k)} + \lambda^* s^{(k)} \quad (4.20)$$

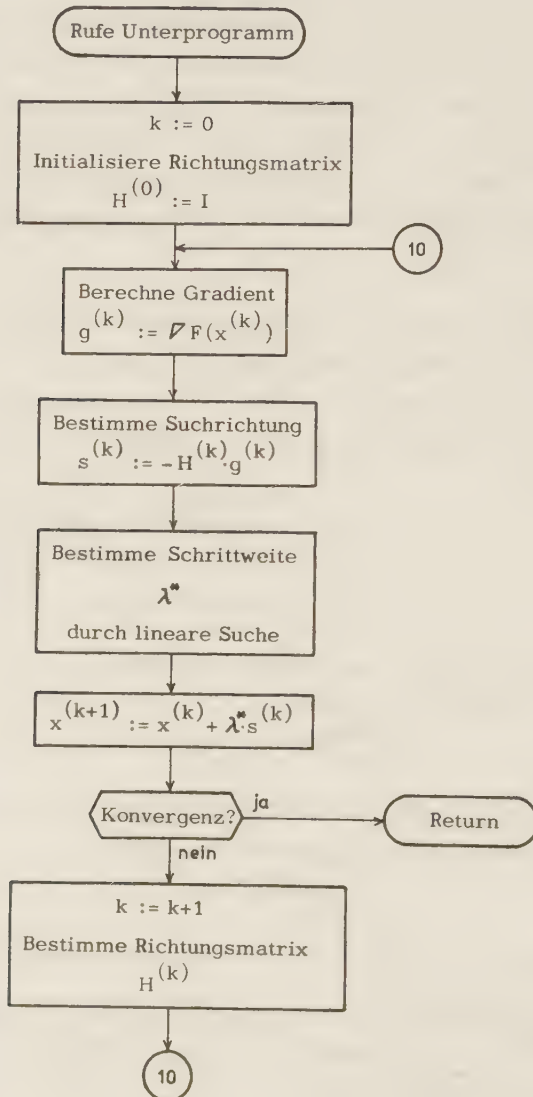
wobei der Skalar λ^* die Schrittweite angibt. λ^* wird durch minimieren der Funktion einer einzigen Variablen

$$\hat{F}(\lambda) := F(x^{(k)} + \lambda \cdot s^{(k)}) \quad (4.21)$$

bestimmt. $F(x)$ ist hierbei die zu minimierende Penaltyfunktion.

Eine neue Formel zur iterativen Bestimmung der Richtungsmatrix $H^{(k+1)}$ wurde unabhängig voneinander von BROYDEN [1], FLETCHER [4] und SHANNO [25] entwickelt:

Ablaufplan für unbedingte Minimierung



$$H^{(k+1)} := H^{(k)} - \frac{\Delta x \cdot \Delta g^T \cdot H^{(k)}}{\Delta x^T \cdot \Delta g} - \frac{H^{(k)} \Delta g \cdot \Delta x^T}{\Delta x^T \cdot \Delta g} + \frac{\Delta x \cdot \Delta x^T}{\Delta x^T \cdot \Delta g} + \frac{\Delta g^T \cdot H^{(k)} \cdot \Delta g \cdot \Delta x \cdot \Delta x^T}{(\Delta x^T \cdot \Delta g)^2} ; (k=0,1,2,\dots) \quad (4.22)$$

Die Subroutine NEMT benutzt das oben beschriebene Minimierungsverfahren ohne Nebenbedingungen mit der Broyden-Fletcher-Shanno- (BFS-) Iterationsformel zur Bestimmung der Richtungsmatrix.

Das Verfahren ist in NEMT so implementiert, daß es bei Erfüllung eines der folgenden Kriterien abbricht:

$$a) |x_j^{(k+1)} - x_j^{(k)}| \leq \epsilon_R |x_j^{(k+1)}| + \epsilon_A \quad (4.23)$$

für alle Komponenten x_j ($j=1,\dots,n$) von x während n aufeinanderfolgende Iterationen. ϵ_R und ϵ_A sind vom Benutzer zu bestimmende kleine positive Werte. (Siehe (4.16) und (4.17)).

$$b) \|g^{(k+1)}\| \leq 10^{-10} \quad (4.24)$$

$$c) \|\Delta x\| \leq 10^{-10} \quad (4.25)$$

$$d) |\Delta g^T \cdot H^{(k)} \cdot \Delta g| < 10^{-20} \quad (4.26)$$

$$e) |\Delta x^T \cdot \Delta g| < 10^{-20} \quad (4.27)$$

Der Punkt $x^{(k+1)} =: c$ wird sodann als Näherungswert für das unbedingte Minimum (=Zentrum) genommen.

Wichtige Teile des in NEMT implementierten Verfahrens, insbesondere auch das Abbruchkriterium a), sind von LOOTSMA's Subroutine MINI [20] übernommen worden. MINI benutzt jedoch anders als NEMT die Formel (4.19) zur Bestimmung der Richtungsmatrix.

4.3 Lineare Suche

Um die Schrittweite λ^* in (4.20) zu bestimmen, ist die Durchführung

einer eindimensionalen Minimierung, genannt lineare Suche, notwendig. Es sei $F(x)$ die Penaltyfunktion (4.8) bzw. (4.13). Gesucht ist das Minimum der Funktion $\hat{F}(\lambda) : \mathbb{R} \rightarrow \mathbb{R}$, definiert durch

$$\hat{F}(\lambda) := F(x^{(k)} + \lambda s^{(k)}) \quad (4.28)$$

wobei $x^{(k)}, s^{(k)} \in \mathbb{R}^n$ fest sind.

Die Subroutine NEMT benutzt für die lineare Suche eine Kombination aus dem Algorithmus von Davies, Swann und Campey und Powells Algorithmus (beide beschrieben in HIMMELBLAU [7]). Das Verfahren ist wie folgt: Es sei

$$\left. \begin{aligned} \lambda_0 &:= 0 \\ \lambda_1 &:= \text{Min}[1, 2 \cdot \lceil \hat{F}(\lambda_0)/d \rceil] \\ \lambda_r &:= 2 \cdot \lambda_{r-1} \quad ; \quad (r = 2, 3, \dots) \end{aligned} \right\} \quad (4.29)$$

wobei $d := s^{(k)T} \cdot g^{(k)} (g^{(k)} := \nabla F(x^{(k)}))$.

Für den Fall, daß

$$\hat{F}(\lambda_1) \leq \hat{F}(\lambda_0) \text{ ist,} \quad (4.30)$$

werden $\hat{F}(\lambda_1 + \lambda_2)$, $\hat{F}(\lambda_1 + \lambda_2 + \lambda_3)$, ... berechnet, bis ein $m \in \{2, 3, 4, \dots\}$ gefunden ist mit

$$\hat{F}\left(\sum_{r=1}^m \lambda_r\right) > \hat{F}\left(\sum_{r=1}^{m-1} \lambda_r\right) \quad (4.31)$$

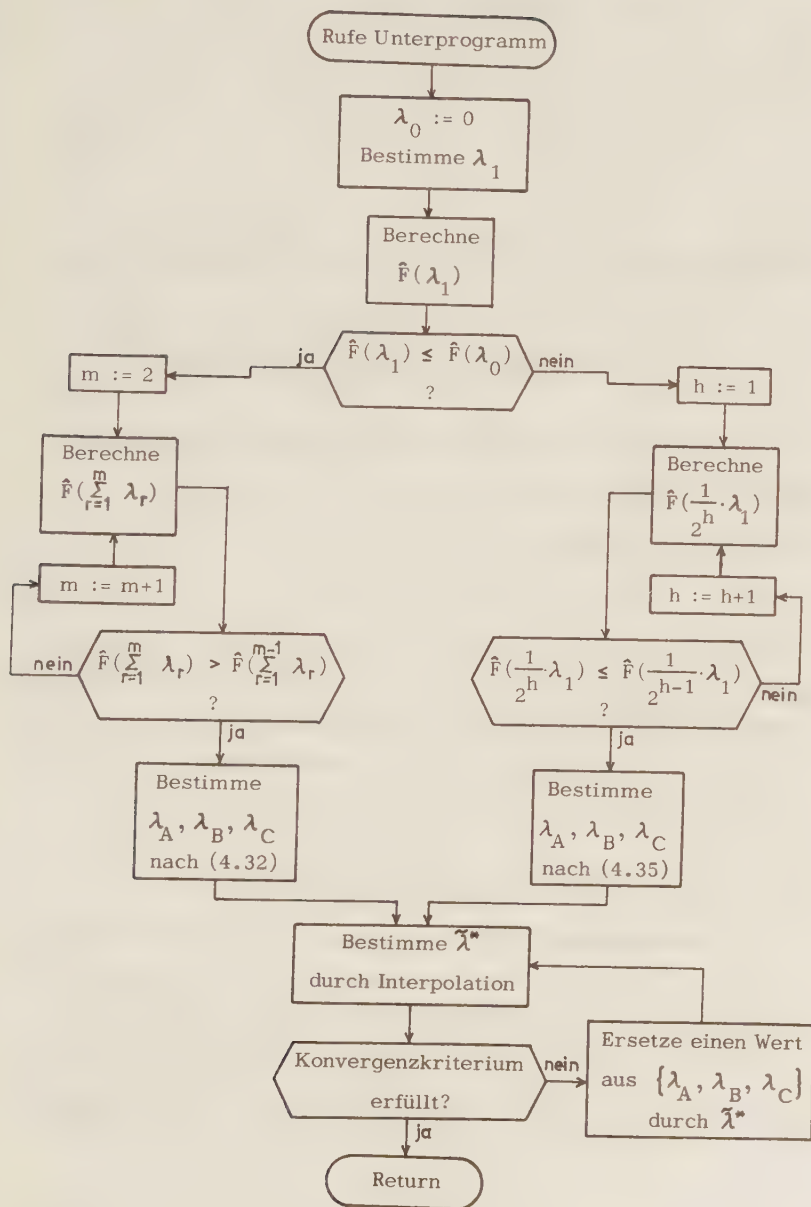
Es werden nun $\lambda_A, \lambda_B, \lambda_C$ definiert durch

$$\left. \begin{aligned} \lambda_A &:= \sum_{r=1}^{m-2} \lambda_r \\ \lambda_B &:= \sum_{r=1}^{m-1} \lambda_r \\ \text{und } \lambda_C &:= \sum_{r=1}^m \lambda_r \end{aligned} \right\} \quad (4.32)$$

Gilt statt (4.30)

$$\hat{F}(\lambda_1) > \hat{F}(\lambda_0) \quad (4.33)$$

Ablaufplan für lineare Suche



so werden $\hat{F}(\lambda_1/2)$, $\hat{F}(\lambda_1/4)$, ... berechnet, bis ein $h \in \{1, 2, 3, \dots\}$ gefunden ist mit

$$\hat{F}(\lambda_1/2^h) \leq \hat{F}(\lambda_0) \quad (4.34)$$

In diesem Fall werden λ_A , λ_B , λ_C definiert durch

$$\left. \begin{aligned} \lambda_A &:= \lambda_0 = 0 \\ \lambda_B &:= \lambda_1/2^h \\ \text{und } \lambda_C &:= \lambda_1/2^{h-1} \end{aligned} \right\} \quad (4.35)$$

In beiden möglichen Fällen wird auf λ_A , λ_B und λ_C interpoliert durch

$$\tilde{\lambda}^* := \frac{1}{2} \frac{(\lambda_C^2 - \lambda_A^2) \cdot \hat{F}(\lambda_B) + (\lambda_A^2 - \lambda_B^2) \cdot \hat{F}(\lambda_C) + (\lambda_B^2 - \lambda_C^2) \cdot \hat{F}(\lambda_A)}{(\lambda_B - \lambda_C) \cdot \hat{F}(\lambda_A) + (\lambda_C - \lambda_A) \cdot \hat{F}(\lambda_B) + (\lambda_A - \lambda_B) \cdot \hat{F}(\lambda_C)} \quad (4.36)$$

(4.36) erhält man durch ableiten und Null setzen des Lagrangeschen Interpolationspolynoms [2] .

Erfüllt $\tilde{\lambda}^*$ die Bedingung

$$^8 \quad |\tilde{\lambda}^* - \lambda_B| \leq \epsilon_R |\tilde{\lambda}^*| + \epsilon_A \quad (4.37)$$

wobei ϵ_R und ϵ_A wie in (4.23) bzw. wie in (4.16) und (4.17) sind, so wird $\tilde{\lambda}^*$ als ausreichend genaue Approximation des Minimums von $\hat{F}(\lambda)$ genommen, und $\lambda^* := \tilde{\lambda}^*$ ist die gesuchte Schrittweite. Andernfalls wird der ungünstigste Wert aus $\{\lambda_A, \lambda_B, \lambda_C\}$ durch $\tilde{\lambda}^*$ ersetzt und die Interpolation wiederholt.

4.4 Numerische Berechnung der Ableitungen

Durch Parametereinstellung wird der Subroutine NEMT beim Aufruf mitgeteilt, ob die Ableitungen der Problemfunktionen für die Berechnung des Gradienten der Penaltyfunktion analytisch vorgegeben sind oder nicht. Im letzteren Fall werden diese Ableitungen vom Verfahren selbst numerisch berechnet. Der Benutzer kann dabei zwischen den beiden folgenden

Approximationsformeln wählen:

$$\frac{\partial f_i(x)}{\partial x_j} \approx \frac{f_i(x + \delta_j e_j) - f_i(x - \delta_j e_j)}{2 \delta_j} \quad (4.38)$$

$$\frac{\partial f_i(x)}{\partial x_j} \approx \left[4 \frac{f_i(x + \frac{1}{2} \delta_j e_j) - f_i(x - \frac{1}{2} \delta_j e_j)}{\delta_j} - \frac{f_i(x + \delta_j e_j) - f_i(x - \delta_j e_j)}{2 \delta_j} \right] / 3 \quad (4.39)$$

Dabei gilt: $f_i(x)$ ($i=1, \dots, m+1$) sind die Zielfunktion und die m Nebenbedingungen; x_j ($j=1, \dots, n$) sind die Komponenten von $x \in \mathbb{R}^n$; e_j ($j=1, \dots, n$) sind die Einheitsvektoren; δ_j ($j=1, \dots, n$) sind vom Programm durch

$$\delta_j := 10^{-3} \cdot |x_j| + 10^{-6} \quad (4.40)$$

bestimmte kleine Werte (Schrittweiten).

Die Formeln (4.38) und (4.39) entsprechen den Anfangseintragungen des Romberg-Schemas [9] für die numerische Berechnung von Ableitungen.

Für die meisten praktischen Aufgaben ist die Approximation der Ableitungen durch (4.38) genügend genau. Da bei Verwendung von (4.38) nur zwei Funktionsaufrufe pro partieller Ableitung benötigt werden, gegenüber vier bei Verwendung von (4.39), ist erstere Formel vorzuziehen. Der Benutzer gibt die gewünschte Approximationsformel durch Parametereinstellung beim Aufruf der Subroutine NEMT bekannt. Das oben beschriebene, in NEMT implementierte Verfahren zur numerischen Berechnung der Ableitungen ist der Subroutine MINI von LOOTSMA [20] entnommen.

4.5 Extrapolation gegen das Minimum des folgenden Minimierungszyklus ohne Nebenbedingungen

Nach Beendigung von k unbedingten Minimierungszyklen versucht NEMT

gegen das Zentrum c_{k+1} des $(k+1)$ -ten Zyklus zu extrapolieren. Hierzu wird ein quadratisches Polynom verwendet:

$$\bar{x}_j^{(k+1)} \approx \tilde{x}_j^{(k+1)} := A_j + B_j \cdot t_{k+1} + C_j \cdot t_{k+1}^2 \quad ; \quad (j=1, \dots, n) \quad (4.41)$$

Es ist dabei $c_{k+1}^T = [\bar{x}_1^{(k+1)}, \dots, \bar{x}_n^{(k+1)}]$ und $\tilde{c}_{k+1}^T = [\tilde{x}_1^{(k+1)}, \dots,$

$\dots, \tilde{x}_n^{(k+1)}]$. Die Koeffizienten des Polynoms, A_j , B_j und C_j ($j=1, \dots, n$),

werden durch Lösen der folgenden n Gleichungssysteme (ein Gleichungssystem für jede Koordinate in $\mathbb{R}^n$) errechnet, die man durch Einsetzen schon bekannter Werte in (4.41) erhält:

$$\left. \begin{aligned} \bar{x}_j^{(k-2)} &= A_j + B_j \cdot t_{k-2} + C_j \cdot t_{k-2}^2 \\ \bar{x}_j^{(k-1)} &= A_j + B_j \cdot t_{k-1} + C_j \cdot t_{k-1}^2 \\ \bar{x}_j^{(k)} &= A_j + B_j \cdot t_k + C_j \cdot t_k^2 \end{aligned} \right\} \quad ; \quad (j=1, \dots, n) \quad (4.42)$$

Sei M die Koeffizientenmatrix von (4.42),

$$M := \begin{pmatrix} 1 & t_{k-2} & t_{k-2}^2 \\ 1 & t_{k-1} & t_{k-1}^2 \\ 1 & t_k & t_k^2 \end{pmatrix} \quad (4.43)$$

dann läßt sich (4.42) wie folgt schreiben:

$$\begin{pmatrix} \bar{x}_j^{(k-2)} \\ \bar{x}_j^{(k-1)} \\ \bar{x}_j^{(k)} \end{pmatrix} = M \cdot \begin{pmatrix} A_j \\ B_j \\ C_j \end{pmatrix} \quad ; \quad (j=1, \dots, n) \quad (4.44)$$

Daraus ergibt sich für die Koeffizienten des Polynoms:

$$\begin{pmatrix} A_j \\ B_j \\ C_j \end{pmatrix} = M^{-1} \cdot \begin{pmatrix} \bar{x}_j^{(k-2)} \\ \bar{x}_j^{(k-1)} \\ \bar{x}_j^{(k)} \end{pmatrix} ; (j=1, \dots, n) \quad (4.45)$$

M^{-1} existiert, da wegen (3.28) bzw. (4.4) $t_k > t_{k-1} > t_{k-2}$ (vorausgesetzt die optimale zulässige Lösung ist noch nicht erreicht) und folglich $\det M \neq 0$ ist (det M ist die Vandermondesche Determinante). Falls M schlecht konditioniert ist, wird die Extrapolation abgebrochen.

Durch Einsetzen der neuen Stutzebene t_{k+1} sowie der aus (4.45) errechneten Koeffizienten A_j , B_j und C_j in das Polynom (4.41) kann nun $\tilde{x}_j^{(k+1)}$ bestimmt werden.

Im Fall, daß erst zwei unbedingte Minimierungszyklen beendet worden sind, wird anstelle von (4.41) ein lineares Polynom benutzt:

$$\tilde{x}_j^{(k+1)} := A_j + B_j \cdot t_{k+1} \quad (4.46)$$

Die Extrapolation wird als erfolgreich angesehen, wenn der extrapolierte Punkt $\tilde{c}_{k+1}$ einen niedrigeren Wert für die Penaltyfunktion ergibt als c_k :

$$F_{k+1}(\tilde{c}_{k+1}) \leq F_{k+1}(c_k) \quad (4.47)$$

In diesem Fall wird im folgenden unbedingten Minimierungszyklus (mit t_{k+1} als Stutzebene) $\tilde{c}_{k+1}$ als Anfangspunkt genommen.

V. PRAKTISCHE ERFAHRUNGEN

Die Subroutine NEMT wurde anhand von 12 Aufgaben getestet und mit der Subroutine MINI von LOOTSMA [20] verglichen. Die Testaufgaben sind HIMMELBLAU [7] entnommen und in Anhang B dieser Arbeit wiedergegeben. Alle Rechnungen wurden an der TR 440 am "Großrechenzentrum für die Wissenschaft in Berlin", 1 Berlin 31, Heilbronner Str.10, durchgeführt.

5.1 Bemerkungen zu NEMT

Die Subroutine NEMT kann sowohl Aufgaben mit vorgegebenen analytischen Ableitungen minimieren, als auch bei Aufgaben, wo die Ableitungen nicht analytisch vorgegeben sind, diese numerisch berechnen (siehe Kapitel 4.4). Ob die Ableitungen numerisch berechnet werden müssen oder nicht, wird NEMT durch Parametereinstellung bekanntgegeben. Durch Parametereinstellung kann auch bestimmt werden, ob NEMT die normierte quadratische Loss-Funktion minimiert, oder die gewöhnliche quadratische Loss-Funktion mit veränderlichen Stützungen (siehe Kapitel 4.1). Es ist auch möglich, durch Parametereinstellung zu erreichen, daß nur die Nebenbedingungen "normiert" werden, bzw. nur die Zielfunktion.

Bei der Mehrheit der Aufgaben wird die Konvergenz der Folge der Stützebenen durch die Normierungen erheblich beschleunigt und damit die Rechenzeit verkürzt. Wie das in Tabelle 5.1 angegebene Beispiel zeigt, wird eine Konvergenzverbesserung sowohl durch Normieren der Nebenbedingungen, als auch durch Normieren der Zielfunktion, erreicht.

Auf den Begriff "äquivalente Funktionsaufrufe" wird in Kapitel 5.3 näher eingegangen. Bezüglich weiterer einstellbarer Parameter der Subroutine NEMT wird auf die Anleitung zur Benutzung von NEMT, bzw. auf die

Comment-Karten des Programms in Anhang C verwiesen. Soweit bei den Rechnungen andere als in Anhang C empfohlene Parameterwerte benutzt wurden, wird dies im folgenden ausdrücklich betont.

NEMT	Rechenzeit in Sekunden	Äquivalente Funktionsaufrufe
ohne Normierungen	95.84	4855
mit Normierung der Zielfunktion	79.50	3983
mit Normierung der Nebenbedingungen	76.00	4023
mit Normierung der Zielfunktion und Nebenbedingungen	58.29	2989

Tabelle 5.1 Vergleich von NEMT mit und ohne Normierungen anhand von Aufgabe 11.

5.2 Beschreibung des Testverfahrens

Die Testaufgaben wurden jeweils für drei verschiedene Parametereinstellungen von der Subroutine NEMT gerechnet:

1. Mit Normierungen und analytisch vorgegebenen Ableitungen, im folgenden mit NEMT-A bezeichnet.
2. Mit Normierungen und numerisch zu berechnenden Ableitungen, im folgenden mit NEMT-N bezeichnet.
3. Ohne Normierungen (also die gewöhnliche quadratische Loss-Funktion mit veränderlichen Stützungen) und mit analytisch vorgegebenen Ableitungen, im folgenden mit EMT-A bezeichnet.

Um eine Vergleichsmöglichkeit zu bestehenden Optimierungsverfahren

auf Penaltyfunktionsbasis zu haben, wurden die Testaufgaben außerdem von LOOTSMA's Subroutine MINI gerechnet. MINI benutzt die gemischte parametrische Penaltyfunktion (2.10). Die Subroutine MINI wurde gewählt, da diese der Subroutine NEMT sehr ähnlich ist (Teile aus MINI wurden in NEMT übernommen) und deshalb besonders zum Vergleich geeignet ist. Zum Beispiel benutzen MINI und NEMT sehr ähnliche Konvergenzkriterien und liefern ähnlichen Output. Für das Testverfahren wurde sowohl MINI mit analytisch vorgegebenen Ableitungen verwendet, im folgenden mit MINI-A bezeichnet, als auch mit numerisch zu berechnenden Ableitungen, im folgenden mit MINI-N bezeichnet.

Die Numerierung der Testaufgaben stimmt nicht mit der Numerierung in HIMMELBLAU [7] überein. (In Anhang B sind die in HIMMELBLAU verwendeten Nummern in Klammern angegeben). Die hier verwendete Numerierung gibt in etwa den Schwierigkeitsgrad der Aufgaben wieder, d. h. mit größer werdender Nummer nimmt auch der Schwierigkeitsgrad zu (wobei Schwierigkeitsgrad jedoch ein subjektiver, vom verwendeten Optimierungsverfahren abhängiger Begriff ist). Tabelle 5.2 gibt einen Überblick über die Eigenschaften der gerechneten Aufgaben. Schranken auf den unabhängigen Variablen (obere oder untere Grenzen) sind in Tabelle 5.2 in der Spalte "lineare Ungleichungen" mit eingeschlossen, da sie von NEMT als solche behandelt werden. MINI hingegen bietet die Möglichkeit, Nichtnegativitätsbedingungen der Variablen gesondert zu behandeln, welches eine geringfügige Entlastung beim Codieren der Aufgaben und des aufrufenden Programms bedeutet. Bei allen Rechnungen wurde bei den entsprechenden Aufgaben von dieser Möglichkeit bei MINI Gebrauch gemacht.

Aufgaben Nr.	Anzahl der Nebenbedingungen				Anzahl der Variab- len
	lineare Un- gleichungen	lineare Gleichungen	nicht- lineare Un- gleichungen	nichtlineare Gleichungen	
1	-	-	-	-	2
2	1 (1)	-	1 (1)	-	2
3	-	1	1 (1)	-	2
4	3 (0)	1	-	1	3
5	4 (2)	-	3 (0)	-	2
6	10 (3)	-	6 (2)	-	5
7	6 (1)	-	14 (1)	-	3
8	12 (5)	-	4 (1)	-	6
9	1 (1)	-	13 (8)	-	9
10	10 (0)	3	-	-	10
11	-	-	-	3	10
12	32 (5)	8	-	-	16

Tabelle 5.2 Anzahl der Variablen und der Nebenbedingungen der Testaufgaben. Die Zahlen in Klammern geben die Anzahl der in der optimalen zulässigen Lösung aktiven Nebenbedingungen an.

5.3 Ergebnisse

Die Testergebnisse sollen anhand der folgenden Tabellen besprochen werden. Für den Vergleich wurden bei jeder Rechnung die benötigte Netto-Rechenzeit sowie die Anzahl der "äquivalenten Funktionsaufrufe" vermerkt.

Die Netto-Rechenzeit wurde bestimmt, indem jeweils vor und nach Aufruf von NEMT bzw. MINI die Bibliothekssubroutine TIME aufgerufen wurde. Die Anzahl der Äquivalenten Funktionsaufrufe EF berechnet sich wie folgt:

$$EF = FL + FN + n \cdot D \quad (5.1)$$

Dabei bezeichnet FL die Anzahl der Funktionsberechnungen für die lineare Suche, FN die Anzahl der Funktionsberechnungen für die numerische Berechnung der Ableitungen, und D bezeichnet die Anzahl der Berechnungen der analytisch vorgegebenen Ableitungen (n ist die Anzahl der unabhängigen Variablen). Es sei hier darauf hingewiesen, daß sowohl NEMT-A als auch MINI-A die Ableitungen am Anfang einmal numerisch berechnen, damit der Benutzer diese mit seinen analytisch vorgegebenen Ableitungen vergleichen kann. Dies hat sich als äußerst nützlich erwiesen, um eventuelle Fehler in den analytisch berechneten Ableitungen zu entdecken.

In Tabelle 5.3 sind die jeweils benötigten Rechenzeiten zur Lösung der Testaufgaben angegeben. Aufgabe 6 und Aufgabe 12 wurden mit je zwei verschiedenen Startpunkten gerechnet, sie sind mit 6(a) und 6(b) bzw. 12(a) und 12(b) bezeichnet. Aufgabe 8 wurde in zwei Versionen gerechnet, die sich in vier Konstanten unterscheiden. Sie sind mit 8-1 und 8-2 bezeichnet. Aufgabe 7 wurde nur mit numerisch berechneten Ableitungen gerechnet, da diese Aufgabe in Form einer FORTRAN-Routine gegeben ist, und der Aufwand für die analytische Berechnung der Ableitungen nicht gerechtfertigt schien.

Tabelle 5.4 gibt die Anzahl der äquivalenten Funktionsaufrufe an. Ein Vergleich mit Tabelle 5.3 zeigt, daß höhere Anzahl von Funktionsaufrufen nicht immer längere Rechenzeit bedeutet. Im allgemeinen ist die Anzahl der äquivalenten Funktionsaufrufe ein ungenaues Vergleichskriterium

Aufgaben Nr.	NEMT-A	NEMT-N	EMT-A	MINI-A	MINI-N
1	0.91	0.98	0.91	1.04	1.26
2	1.30	1.45	2.06	1.55	1.64
3	1.24	1.27	2.89	1.80	1.92
4	1.33	1.52	2.83	2.07	2.62
5	¹⁾ 2.28	¹⁾ 2.72	¹⁾ 2.68	¹⁾ 2.53	¹⁾ 2.84
6(a)	8.93	11.14	17.88	6.14	7.97
6(b)	8.10	10.91	20.16	19.06	22.70
7	-	12.40	²⁾ 12.49	-	28.16
8-1	12.44	15.95	13.47	5.90	8.36
8-2	17.25	24.05	22.66	12.91	18.15
9	25.84	42.42	10.52	24.53	40.53
10	14.47	28.81	33.37	23.91	47.47
11	58.29	123.49	95.84	58.58	183.57
12(a)	207.63	1158.39	357.76	353.31	1814.26
12(b)	205.00	1109.17	358.64	199.60	1003.62

Tabelle 5.3 Rechenzeiten in Sekunden.

¹⁾ Es wurde ein lokales, aber nicht das globale Optimum erreicht. ²⁾ Mit numerisch berechneten Ableitungen.

Aufgaben Nr.	NEMT-A	NEMT-N	EMT-A	MINI-A	MINI-N
1	193	267	193	221	323
2	309	361	315	248	298
3	176	226	332	219	273
4	206	278	425	338	453
5	1) 358	1) 432	1) 403	1) 250	1) 340
6(a)	1760	2101	2887	906	1281
6(b)	1468	2134	3374	2486	3337
7	-	718	2) 746	-	1393
8-1	2079	2736	2359	724	1144
8-2	3043	4478	3772	1837	2793
9	4341	6214	1650	3353	5342
10	1538	2442	3451	2418	4054
11	2989	5051	4855	2994	7450
12(a)	5206	9482	8962	8704	15659
12(b)	5121	9061	8767	4664	8204

Tabelle 5.4 Äquivalente Funktionsaufrufe.

1) und 2) siehe Tabelle 5.3.

für Algorithmen, bietet aber den Vorteil, im Gegensatz zur Rechenzeit, relativ unabhängig von der Hardware zu sein. Wegen ihrer strukturellen Ähnlichkeit dürfte im Vergleich von NEMT und MINI die Anzahl der äquivalenten Funktionsaufrufe jedoch ein recht brauchbares Vergleichskriterium sein, trotz der Diskrepanz zwischen Tabelle 5.3 und Tabelle 5.4.

Aufgabe 1 hat keine Nebenbedingungen (siehe Tabelle 5.2), und ihre Minimierung beschränkt sich auf ein einmaliges Anwenden des BFS-Algorithmus bzw. des DFP-Algorithmus (siehe Kapitel 4.2). Aufgabe 1 ist deshalb für das Testen von Penaltyfunktionsverfahren wenig geeignet, wurde aber aus Gründen der Vollständigkeit mit in das Vergleichsverfahren hineingenommen.

Aufgaben 6 und 12 zeigen, daß MINI sehr viel stärker vom gewählten Startpunkt abhängt als NEMT. (Die benutzten Startpunkte, sowie die optimalen Lösungen, sind in Anhang B bei den Testaufgaben angegeben). Dies läßt sich dadurch erklären, daß einmal erfüllte Nebenbedingungen bei MINI während der gesamten Rechnung erfüllt bleiben. Da bei Aufgabe 6(a) ein zulässiger Startpunkt benutzt wurde, reduziert sich MINI in diesem Fall auf die Anwendung der logarithmischen Barriere-Funktion (vergleiche (2.3) und (2.10)).

Am schlechtesten im Vergleich zu MINI schnitt NEMT bei Aufgabe 8 ab. Während bei allen anderen Aufgaben für den ersten unbedingten Minimierungszyklus, sowohl bei NEMT als auch bei MINI, ein Parameter $R=0.05$ verwendet wurde (siehe (2.10) und (4.8)), divergierte NEMT für diesen Parameterwert bei Aufgabe 8. Für diese Aufgabe wurde deshalb $R=0.0005$ in NEMT gesetzt. MINI wurde in diesem Fall mit einem Anfangswert $R=0.01$ gerechnet (dies ist der kleinste einstellbare Anfangswert für R in MINI).

Insgesamt löste NEMT-A 9 Aufgaben schneller als MINI-A (bezüglich Rechenzeit), 5 Aufgaben wurden von MINI-A schneller gelöst. Nimmt man die Anzahl der äquivalenten Funktionsaufrufe als Vergleichsbasis, so erhält man ein Verhältnis 7:7, also ausgeglichen. NEMT-N vergleicht sich mit MINI-N in den gleichen Verhältnissen, wenn man Aufgabe 7 nicht mitzählt, die von NEMT-N schneller gelöst wurde.

EMT-A ist NEMT-A fast durchweg unterlegen. Nur Aufgabe 9 wurde eindeutig von EMT-A am schnellsten gelöst, bezüglich beider Vergleichskriterien. In Aufgabe 9 haben sich also die Normierungen als nachteilig gezeigt. Zählt man die Erfolge von NEMT-A und EMT-A zusammen, erhält man ein Verhältnis von 10:4 (bzgl. Rechenzeit) bzw. 8:6 (bzgl. Funktionsaufrufe) zugunsten von NEMT im Vergleich mit MINI.

STAHA [26] schlägt zum Vergleich von Algorithmen eine "skalierte Rechenzeit" vor, definiert durch

$$T_{s_i} := 100 \cdot T_{\min} / T_i \quad (5.2)$$

dabei ist T_{s_i} die skalierte Rechenzeit für den i-ten Algorithmus. T_i ist die tatsächliche Rechenzeit des i-ten Algorithmus (bei einer bestimmten Testaufgabe) und $T_{\min}$ ist die kürzeste Rechenzeit, in der die Aufgabe gelöst wurde (von einem der zu vergleichenden Algorithmen). Diese skalierte Rechenzeit wurde für alle Testaufgaben berechnet und die Durchschnittswerte gebildet. (Aufgaben 6, 8 und 12 wurden bei der Bildung der Durchschnittswerte jeweils als einzelne Aufgabe aufgefaßt, d. h., es wurde zuvor jeweils das Mittel aus den Ergebnissen von 6(a) und 6(b), 8-1 und 8-2 sowie 12(a) und 12(b) gebildet.)

Auf analoge Weise wurde bei den äquivalenten Funktionsaufrufen vorgegangen und durchschnittliche, skalierte äquivalente Funktionsaufrufe gebildet. Nach den so erhaltenen Werten können die benutzten Optimierungs-

verfahren miteinander verglichen und gewertet werden.

Die skalierten Ergebnisse sind in Tabelle 5.5 wiedergegeben. Bei der Berechnung dieser Werte wurde Aufgabe 7 nicht berücksichtigt, da die Ergebnisse bei dieser Aufgabe mit analytisch vorgegebenen Ableitungen fehlen. Aus Tabelle 5.5 geht eindeutig die Überlegenheit der normierten Loss-Funktion gegenüber der gewöhnlichen Loss-Funktion mit veränderlichen Stützungen hervor. Die erste Spalte von Tabelle 5.5 läßt auch auf eine Überlegenheit von NEMT gegenüber MINI schließen, der Unterschied in der zweiten Spalte ist jedoch nicht signifikant.

	durchschnittliche skalierte Rechenzeit	durchschnittliche skalierte äquiv. Funktionsaufrufe
NEMT-A	89.54	82.45
MINI-A	77.11	81.89
EMT-A	62.43	61.92
NEMT-N	63.79	58.38
MINI-N	52.56	54.76

Tabelle 5.5 Skalierte Daten.

5.4 Broyden-Fletcher-Shanno versus Davidon-Fletcher-Powell

Während der Durchführung des Testverfahrens, nachdem schon ein Teil der Testaufgabe gerechnet worden war und ein Vergleich NEMT gegenüber MINI bevorzugte, kam der Verdacht auf, daß dies möglicherweise nicht so sehr durch die unterschiedlichen Penaltyfunktionen bedingt sei, auf die NEMT und MINI beruhen, sondern vielmehr durch die unterschiedlichen Iterationsformeln zur Bestimmung der Richtungsmatrix in den unbedingten Minimierungszyklen (siehe Kapitel 4.2). Aus diesem

Grund wurden die restlichen Testaufgaben zusätzlich von einer abgeänderten Version von MINI gerechnet, bei der die Richtungsmatrix nicht durch die DFP-Formel (4.19) bestimmt wird, sondern durch die BFS-Formel (4.22). Diese Version von MINI ist im folgenden mit MINI-(BFS)-A bezeichnet. Die Ergebnisse von MINI(BFS)-A sind in Tabelle 5.6 mit den Ergebnissen von MINI-A verglichen.

Aufgabe	Rechenzeit		Äquivalente Funktionsaufrufe	
	MINI(BFS)-A	MINI-A	MINI(BFS)-A	MINI-A
2	1.44	1.55	259	248
4	2.38	2.07	331	338
5	1) 2.43	1) 2.53	1) 243	1) 250
6(a)	7.04	6.14	933	906
6(b)	17.87	19.06	2276	2486
7	2) 13.15	2) 28.16	2) 616	2) 1393
8-1	6.53	5.90	724	724
9	24.63	24.53	2981	3353
12(a)	338.47	353.31	8108	8704
12(b)	217.81	199.60	4755	4664

Tabelle 5.6 Vergleich BFS – DFP.

1) Es wurde ein lokales, aber nicht das globale Optimum erreicht. 2) Mit numerisch berechneten Ableitungen.

Wider Erwarten zeigt Tabelle 5.6 ziemliche Ausgeglichenheit im Vergleich der beiden Verfahren; nur bei Aufgabe 7 ist das Ergebnis wirklich signifikant unterschiedlich (Aufgabe 7 wurde bei der Berechnung der Tabelle 5.5 nicht berücksichtigt). Dies mag an dem besonderen Charakter

von Aufgabe 7 liegen.

Aus Tabelle 5.6 errechnet sich (ohne Aufgabe 7) die durchschnittliche skalierte Rechenzeit für MINI-A zu 97.68 und für MINI(BFS)-A zu 95.19. Die durchschnittlichen skalierten äquivalenten Funktionsaufrufe ergeben 96.63 für MINI-A und 99.05 für MINI(BFS)-A. Daraus läßt sich der Stellenwert von MINI(BFS)-A in Tabelle 5.5 projizieren. Die projizierte durchschnittliche skalierte Rechenzeit von MINI(BFS)-A für Tabelle 5.5 ist $77.11 \times (95.19/97.68) = 75.14$. Bei den durchschnittlichen skalierten äquivalenten Funktionsaufrufen erhält man für Tabelle 5.5 den projizierten Wert $81.89 \times (99.05/96.63) = 83.94$. Ein Grund für das bessere Abschneiden von MINI-A gegenüber MINI(BFS)-A bezüglich der skalierten Rechenzeit ist möglicherweise, daß Formel (4.19) in sich kürzer ist als Formel (4.22).

5.5 Einige zusätzliche Ergebnisse

Es soll hier noch einmal das Verhalten und die Konvergenz der NEMT-Penaltyfunktion (mit Normierungen) mit der gewöhnlichen äußeren Penaltyfunktion mit veränderlichen Stützungen (Verfahren EMT-A) verglichen werden.

Die Extrapolation (siehe Kapitel 4.5) erwies sich bei EMT-A als sehr beschleunigend, hatte aber bei NEMT (mit Normierungen) nur geringen Einfluß. Dies liegt wahrscheinlich daran, daß der Konvergenzwert bei NEMT (mit Normierungen) meistens sowieso schon sehr viel günstiger ist als bei EMT-A. Das zeigt die Tabelle 5.7, in der die Anfangskonvergenzwerte (siehe Kapitel 3.4)

$$C_1 = \frac{t_2 - t_1}{t^* - t_1} \quad (5.3)$$

der beiden Verfahren für die Testaufgaben miteinander verglichen

werden. C_1 ist der Konvergenzwert im ersten t-Zyklus nach Vollendung des parametrischen Zyklus (also im zweiten Zyklus insgesamt).

Aufgaben Nr.	EMT-A	NEMT-A
2	0.53	1
3	0.49	0.99
4	0.47	1
6(a)	0.28	0.86
6(b)	0.28	0.86
8-1	0.52	0.54
8-2	0.15	0.33
9	0.90	1
10	0.51	0.92
11	0.50	0.88
12(a)	0.36	0.64
12(b)	0.36	0.64

Tabelle 5.7 Anfangskonvergenzwerte.

Die Werte in Tabelle 5.7 wurden bestimmt, indem jeweils in Gleichung (5.3) die letzte vom Verfahren bestimmte Stutzebene für t^* eingesetzt wurde, da das jeweils letzte t_k als Näherung derselben angesehen werden kann. Ein Approximieren der Anfangskonvergenzwerte durch Beziehung (3.86) würde geringfügig kleinere Werte ergeben. Bei Aufgaben 1, 5 und 7 konnte C_1 nicht bestimmt werden, da die Verfahren bei diesen Aufgaben keine bzw. nur eine Stutzebene benutzten.

Der Durchschnitt für alle 9 Aufgaben liegt für EMT-A bei 0.49, für NEMT-A bei 0.86. Dies erklärt die bessere Platzierung von NEMT-A vor EMT-A in Tabelle 5.5, obwohl das Verhältnis der Konvergenzwerte nicht dem Verhältnis der durchschnittlichen skalierten Werte aus Tabelle

5.5 entspricht. Die Konvergenzwerte sind einigermaßen proportional der Anzahl der unbedingten Minimierungszyklen, innerhalb dieser Zyklen kann die Anzahl der Iterationen jedoch erheblich variieren. (Und innerhalb der einzelnen Iterationen kann die benötigte Anzahl von Funktionsaufrufen sowie die Rechenzeit variieren). Während jeder Zyklus einem unbedingten Minimum entspricht, bedeutet jede Iteration eine lineare Suche (siehe Kapitel 4.3), d.h., jede Iteration entspricht einer neuen Richtungsmatrix bei der unbedingten Minimierung.

Abbildungen 5.1 bis 5.6 veranschaulichen den Vergleich von NEMT-A, EMT-A und MINI-A anhand von Aufgabe 3 bzw. 12(a). In den graphischen Darstellungen sind jeweils die Werte der Zielfunktion am Ende jedes unbedingten Minimierungszyklus eingetragen. Die Zahlen in runden Klammern geben die Anzahl der im jeweiligen Zyklus benötigten Iterationen an. EMT-A und NEMT-A extrapolieren gegen das Minimum des jeweils folgenden Zyklus und, falls erfolgreich, benutzen den extrapolierten Punkt als Startpunkt im nächsten Zyklus (siehe Kapitel 4.5). Die Abbildungen zeigen, wie auch schon am Anfang dieses Kapitels erwähnt, daß EMT-A sehr viel öfter erfolgreich extrapoliert als NEMT-A. MINI-A extrapoliert nach jedem Zyklus gegen die optimale zulässige Lösung, und MINI-A bricht ab, wenn die Folge der extrapolierten Punkte das Konvergenzkriterium erfüllt. Während also bei MINI die Extrapolation die Konvergenz insgesamt beschleunigt, hat sie bei EMT und NEMT lediglich den Effekt, die benötigte Zahl von Iterationen im folgenden Zyklus zu verringern. Letzterer Effekt wird besonders krass deutlich bei EMT-A, Aufgabe 12(a): nachdem die Zahl der Iterationen in den Zyklen 14 und 15 bis auf 1 gesunken ist, steigt sie, da in Zyklus 15 keine erfolgreiche Extrapolation stattfindet, in Zyklus 16 auf 25.

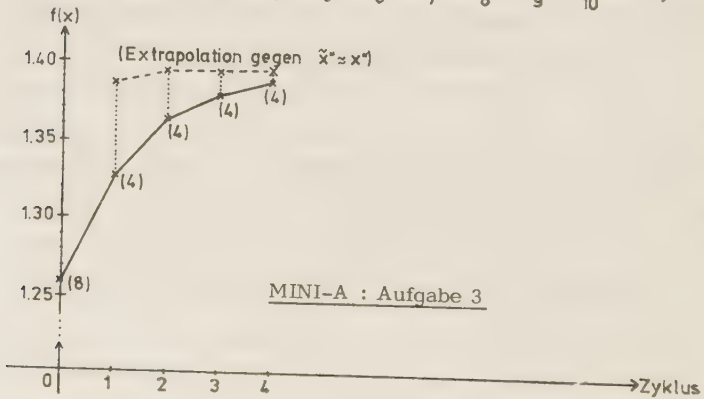
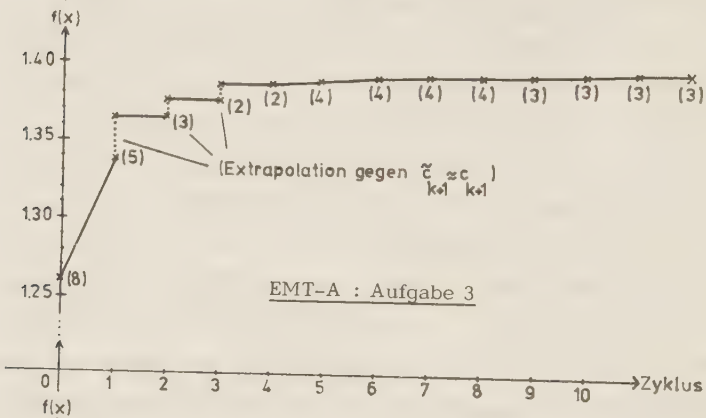
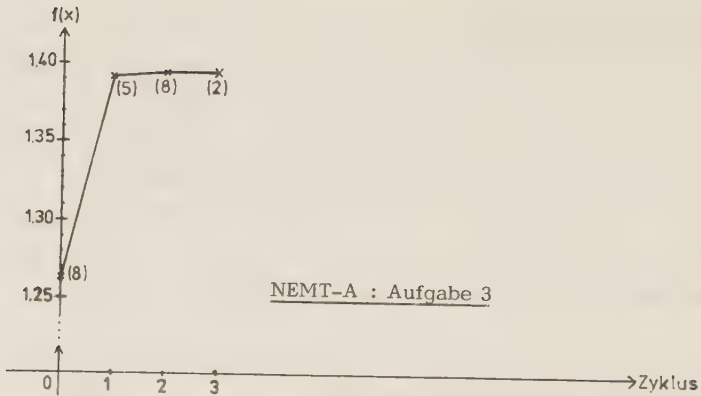


Abb. 5.1-3

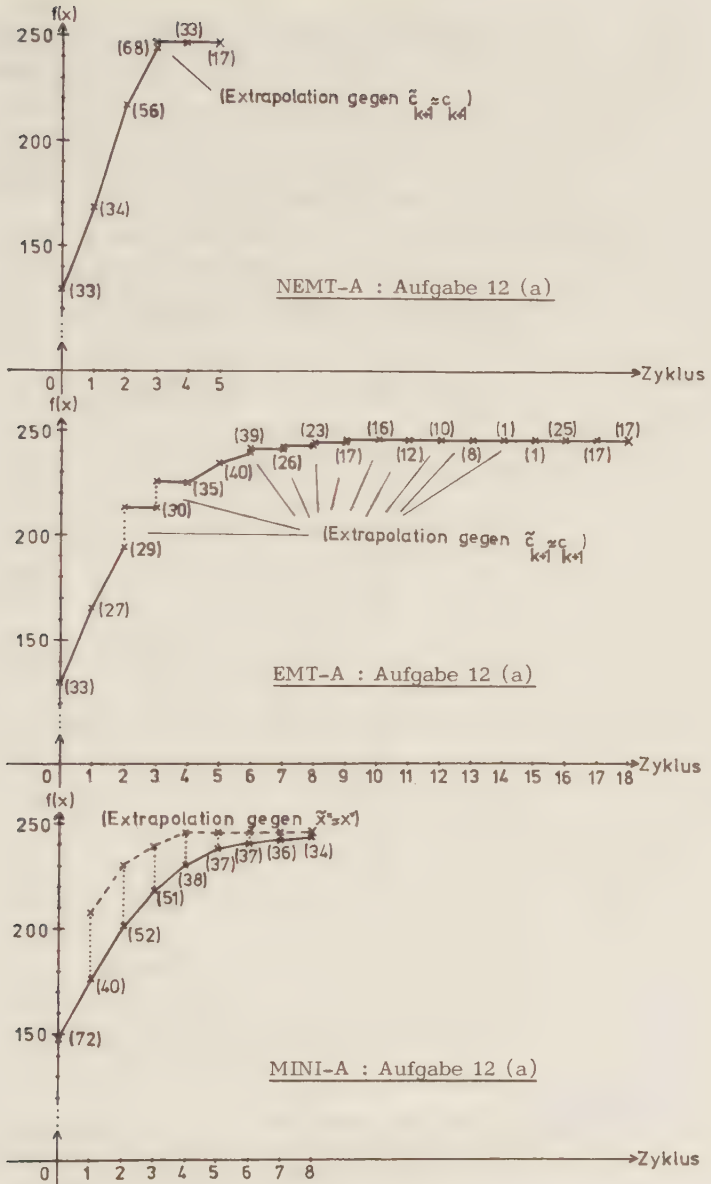


Abb. 5.4-6

Aufgaben Nr.	NEMT-A	EMT-A	MINI-A
1	1 (22)	1 (22)	1 (23)
2	3 (8.3)	9 (4.6)	4 (5.8)
3	4 (5.8)	13 (3.7)	5 (4.8)
4	3 (7.7)	8 (6)	4 (9)
5	2 (11)	2 (14)	4 (6.8)
6(a)	5 (23.4)	17 (11.5)	3 (22)
6(b)	5 (20.6)	17 (13.6)	10 (18.4)
7	1) 2 (20)	1) 2 (21)	1) 3 (33.7)
8-1	9 (11)	10 (12.5)	5 (13.4)
8-2	5 (34.2)	14 (19.2)	6 (21.8)
9	3 (91.7)	4 (26.5)	6 (36.2)
10	5 (19.4)	15 (14.6)	9 (17)
11	5 (40.8)	17 (17.8)	6 (31.2)
12(a)	6 (40.2)	19 (21.4)	9 (44.1)
12(b)	6 (39.2)	19 (20.8)	10 (20.6)

Tabelle 5.8 Anzahl der benötigten Zyklen sowie durchschnittliche Anzahl von Iterationen (in Klammern) pro Zyklus.
1) Mit numerisch berechneten Ableitungen.

In Tabelle 5.8 sind für alle 12 Testaufgaben die jeweils benötigte Anzahl von unbedingten Minimierungszyklen (einschließlich Null-Zyklus) sowie in runden Klammern die durchschnittliche Anzahl von Iterationen pro Zyklus angegeben. Die entsprechenden Werte bezüglich NEMT-N unterscheiden sich nicht wesentlich von den für NEMT-A erhaltenen Werten. Das gleiche gilt für MINI-N und MINI(BFS)-A mit einer Ausnahme: MINI(BFS) benötigte für Aufgabe 7 nur durchschnittlich 14.3 Iterationen pro Zyklus, bei unverändert 3 unbedingten Minimierungszyklen.

SCHLUSSFOLGERUNG

In der vorliegenden Arbeit wurde aus der Loss-Funktion mit veränderlichen Stützungen ein neues Penaltyfunktionsverfahren für die nichtlineare Minimierung mit Nebenbedingungen entwickelt. Die Testergebnisse haben gezeigt, daß dieses neue Verfahren (die normierte Penaltyfunktion) dem gewöhnlichen Loss-Funktionsverfahren mit veränderlichen Stützungen (ohne Normierungen) eindeutig überlegen ist. Die Testergebnisse haben auch gezeigt, daß das neue Verfahren einen Vergleich mit der gemischten parametrischen Penaltyfunktionsmethode standhält. Anders als bei der gemischten Penaltyfunktionsmethode ist das Verhalten des neuen Verfahrens relativ unabhängig vom gewählten Startpunkt. In bestimmten Fällen jedoch ist eine gemischte Penaltyfunktionsmethode einem reinen äußeren Verfahren überlegen (z.B. bei zulässigem Startpunkt oder falls die Zielfunktion nicht im ganzen $\mathbb{R}^n$ definiert ist). Es sollte deshalb weiter untersucht werden, wie die normierte Penaltyfunktion, oder auch die gewöhnliche Loss-Funktion mit veränderlichen Stützungen, doch mit einem inneren Verfahren kombiniert werden könnte.

Eine weitere wichtige Aufgabe für die Forschung ist die Verallgemeinerung von Penaltyfunktionsverfahren auf nichtkonvexe Optimierungsaufgaben. Während in der Praxis Penaltyfunktionen auch bei nichtkonvexen Problemen angewandt werden (mit unterschiedlichem Erfolg), setzen die Konvergenzbeweise die Konvexität (oder zumindest lokale Konvexität) voraus.

LITERATURVERZEICHNIS

- [1] BROYDEN, C.G.: The Convergence of a Class of Double-rank Minimization Algorithms. 1. General Considerations. 2. The New Algorithm.
J. Inst. Maths Applics 6, 1970.
- [2] CARNAHAN, B. / LUTHER, H.A. / WILKES, J.O.: Applied Numerical Methods.
Wiley, New York, 1969.
- [3] FIACCO, A.V. / McCORMICK, G.P.: Nonlinear Programming: Sequential Unconstrained Minimization Techniques.
Wiley, New York, 1968.
- [4] FLETCHER, F.: A New Approach to Variable Metric Algorithms.
Computer J. 13 No. 3, 1970.
- [5] GOFFIN, R.J. / LOOTSMA, F.A.: A Bird's-Eye View of Optimization.
Papers to be presented at a Symposium on Optimization in Eindhoven, 19th February 1974.
N.V. Philips' Gloeilampenfabrieken, Eindhoven 1973.
- [6] HADLEY, G.: Nichtlineare und dynamische Programmierung.
Physica-Verlag, Würzburg - Wien, 1969.
Übersetzung von HADLEY, G.: Nonlinear and Dynamic Programming.
Addison - Wesley, Reading, Mass.
- [7] HIMMELBLAU, D.M.: Applied Nonlinear Programming.
McGraw - Hill, New York, 1972.
- [8] HUARD, P.: Resolution of Mathematical Programming with Non-linear Constraints by the Method of Centres.
In ABADIE, J. (Herausgeber): Nonlinear Programming.
North-Holland-Publ. Co., Amsterdam, 1967.
- [9] JORDAN-ENGELN, G. / REUTTER, F.: Formelsammlung zur Numerischen Mathematik mit Fortran IV - Programmen.
B.I. - Hochschultaschenbücher, Mannheim, 1974.

- [10] KOWALIK, J. / OSBORNE, M.R.: Methods for Unconstrained Optimization Problems.
American Elsevier, New York, 1968.
- [11] KOWALIK, J. / OSBORNE, M.R. / RYAN, D.M.: A New Method for Constrained Optimization Problems.
Oper. Res. 17, 1969.
- [12] KÜNZI, H.P. / ÖTTLI, W.: Nichtlineare Optimierung: Neuere Verfahren Bibliographie.
Springer-Verlag, Berlin, 1969.
- [13] LOOTSMA, F.A.: Logarithmic Programming: A Method for Solving Nonlinear-Programming Problems.
Philips Res. Repts 22, 1967.
- [14] LOOTSMA, F.A.: Constrained Optimization via Penalty Functions.
Philips Res. Repts 23, 1968.
- [15] LOOTSMA, F.A.: Constrained Optimization via Parameter-Free Penalty Functions.
Philips Res. Repts 23, 1968.
- [16] LOOTSMA, F.A.: Boundary Properties of Penalty Functions for Constrained Minimization.
Philips Res. Repts Supplements, No, 3, 1970.
- [17] LOOTSMA, F.A.: A Survey of Methods for Solving Constrained Minimization Problems via Unconstrained Minimization.
In LOOTSMA, F.A. (Herausgeber): Numerical Methods for Nonlinear Optimization.
Academic Press, London, 1972.
- [18] LOOTSMA, F.A.: Penalty-Function Performance of Several Unconstrained-Minimization Techniques.
Philips Res. Repts 27, 1972.
- [19] LOOTSMA, F.A.: Convergence Rates of Quadratic Exterior Penalty-Function Methods for Solving Constrained Minimization Problems.
Paper presented at the Conference on "Numerische Methoden bei Optimierungsaufgaben", Oberwolfach, 1973.

[20] LOOTSMA, F.A.: The Subroutine MINI for Solving Non-linear Optimization Problems.

Philips International Inst. of Technological Studies, Eindhoven, 1974.

[21] LOOTSMA, F.A.: The Design of a Non-linear Optimization Programme for Solving Technological Problems.

Paper presented at the Conference on Optimization Theory and Optimal Control, Oberwolfach, 1974.

[22] PSHENICHNYI, B.N.: Necessary Conditions for an Extremum. Marcel Dekker, New York, 1971.

Übersetzung von PSHENICHNYI, B.N.: Neobkhodimye Usloviya Ekstremuma. Nauka, Moskau, 1969.

[23] SCHENDEL, U.: Numerische Algorithmen bei Optimierungsaufgaben.

Mitteilungen des Bereich Numerische Mathematik und Informatik, Freie Universität Berlin, 1974.

[24] SCHENDEL, U. / WEISTROFFER, R.: Penaltyfunktionsverfahren zur Lösung des nichtlinearen Optimierungsproblems mit Nebenbedingungen. (Preprint).

Mitteilungen des Bereich Numerische Mathematik und Informatik, Freie Universität Berlin, 1975.

[25] SHANNO, D.F.: Conditioning of Quasi-Newton Methods for Function Minimization.

Maths of Comp. 24, 1970.

[26] STAHA, R.L.: Constrained Optimization via Moving Exterior Truncations.

Dissertation, Univ. of Texas at Austin, 1973.

ANHANG A

Beispiel eines
nichtlinearen Optimierungsproblems mit Nebenbedingungen

Die Aufgabe der Bestimmung der chemischen Zusammensetzung eines Gemischs im chemischen Gleichgewicht ist seit langem von Interesse. Solche Aufgaben stellen sich bei der Untersuchung der Leistung von Treibstoffen und in der Synthese komplexer organischer Verbindungen.

Ein Gemisch chemischer Substanzen bei konstanter Temperatur und konstantem Druck erreicht seinen chemischen Gleichgewichtszustand mit der Abnahme der freien Energie des Gemischs auf ein Minimum. Dies folgt aus dem zweiten Hauptsatz der Thermodynamik. Die zu minimierende Zielfunktion bei dem chemischen Gleichgewichtsmodell ist der Ausdruck der freien Energie des betrachteten chemischen Systems. Der Betrag der freien Energie des Gemischs wird minimiert entsprechend den möglichen chemischen Reaktionen zwischen den Komponenten des Gemischs.

Das chemische Gleichgewichtsmodell

Man betrachte ein Gemisch aus m chemischen Elementen. Es sei schon bekannt, daß die m verschiedenen Arten von Atomen chemische Reaktionen eingehen können, um n Verbindungen zu erzeugen, wobei ein einzelnes Atom als mögliche Verbindung angesehen wird.

Man definiere:

x_j := die Anzahl Mole der Verbindung j , die sich im Gleichgewichtszustand im Gemisch befinden.

$$\bar{x} := \sum_{j=1}^n x_j$$

a_{ij} := die Anzahl der Atome des Elements i in einem Molekül der Verbindung j .

b_i := Anteil des Elements i in dem Gemisch.

Für die m Elemente muß jeweils gelten:

$$\sum_{j=1}^n a_{ij} x_j = b_i \quad ; \quad (i=1, \dots, m) \quad (\text{A.1})$$

und

$$x_j \geq 0 \quad ; \quad (j=1, \dots, n) \quad (\text{A.2})$$

Die Bestimmung der Zusammensetzung des Gemischs im Gleichgewicht ist äquivalent zur Bestimmung der Werte von x_j ($j=1, \dots, n$), welche (A.1) erfüllen und außerdem die freie Energie des Systems minimieren. Die Gesamte der freien Energie des Gemischs ist durch

$$\sum_{j=1}^n x_j \left(c_j + \ln \frac{x_j}{x} \right) \quad (\text{A.3})$$

gegeben, mit

$$c_j := \left(\frac{F^0}{RT} \right)_j + \ln p$$

und $(F^0/RT)_j$ entsprechend der Gibbschen freie Energie Funktion unter Standardbedingungen für die j -te Verbindung. (F^0/RT) kann aus thermodynamischen Tabellen entnommen werden. p ist der Gesamtdruck.

Das nichtlineare Optimierungsproblem ist also folgendes: Man suche x_j ($j=1, \dots, n$) welche (A.3) minimieren unter den $m+n$ Nebenbedingungen (A.1) und (A.2).

Anwendungsbeispiel

Es soll die Gleichgewichts-Zusammensetzung bestimmt werden, die entsteht, wenn die Verbindung $\frac{1}{2}\text{N}_2\text{H}_4 + \frac{1}{2}\text{O}_2$ einer Temperatur von 3500°K und einem Druck von 750 psi ($\approx 51\text{ atm}$) ausgesetzt wird. Tabelle A.1 gibt für jede der 10 möglichen Verbindungen (wobei einzelne Atome als Verbindung bezeichnet sind) die Gibbsche freie Energie Funktion $(F^0/RT)_j$, den errechneten Wert von c_j für $p = 750\text{ psi}$ und die Anzahl der Atome von H, N und O pro Molekül. Die Anteile von H, N und O in dem Gemisch sind als $b_1 = 2$, $b_2 = 1$ und $b_3 = 1$ angenommen.

Das sich so ergebende nichtlineare Optimierungsproblem ist die Testaufgabe 10 aus Anhang B.

j	Verbindung	$(F^0/RT)_j$	c_j	a_{ij}		
				i = 1 H	i = 2 N	i = 3 O
1	H	-10.021	-6.089	1		
2	H ₂	-21.096	-17.164	2		
3	H ₂ O	-37.986	-34.054	2		1
4	N	-9.846	-5.914		1	
5	N ₂	-28.653	-24.721		2	
6	NH	-18.918	-14.986	1	1	
7	NO	-28.032	-24.100		1	1
8	O	-14.640	-10.708			1
9	O ₂	-30.594	-26.662			2
10	OH	-26.111	-22.179	1		1

Tabelle A.1

Das obige Beispiel eines nichtlinearen Optimierungsproblems wurde
 "J.BRACKEN und G.P.McCORMICK, Selected Applications of Nonlinear
 Programming, John Wiley & Sons, Inc., New York, 1968" entnommen.

ANHANG B

Testaufgaben

Die Aufgaben sind dem Buch von HIMMELBLAU [7] entnommen, ebenso die Angaben über die Herkunft der Aufgaben. Die Zahlen in Klammern beziehen sich auf die bei HIMMELBLAU verwendete Numerierung.

Aufgabe 1 (2)

H.H.ROSENBROCK, An Automatic Method for Finding the Greatest and Least Value of a Function, Computer J., 3:175 (1960).

Anzahl der Variablen: 2

Anzahl der Nebenbedingungen: 0

Zu minimierende Zielfunktion:

$$f(x) = 100 (x_2 - x_1^2)^2 + (1 - x_1)^2$$

Startpunkt:

$$x^{(0)} = (-1.2, 1)^T$$

$$f(x^{(0)}) = 24.20$$

Ergebnisse:

$$x^* = (1, 1)^T$$

$$f(x^*) = 0$$

Aufgabe 2 (24)

J.BRACKEN und G.P.McCORMICK, Selected Applications of Nonlinear Programming, John Wiley & Sons, Inc., New York, 1968.

Anzahl der Variablen: 2

Anzahl der Nebenbedingungen: 1 nichtlineare Ungleichung
1 lineare Ungleichung

Zu minimierende Zielfunktion:

$$f(x) = (x_1 - 2)^2 + (x_2 - 1)^2$$

Nebenbedingungen:

$$g_1(x) = -x_1^2 + x_2 \geq 0$$

$$g_2(x) = -x_1 - x_2 + 2 \geq 0$$

Nichtzulässiger Startpunkt:

$$\begin{aligned}x^{(0)} &= (2, 2)^T \\ f(x^{(0)}) &= 1 \\ g_1(x^{(0)}) &= -2 \\ g_2(x^{(0)}) &= -2\end{aligned}$$

Ergebnisse:

$$\begin{aligned}x^* &= (1, 1)^T \\ f(x^*) &= 1 \\ g_1(x^*) &= 0 \\ g_2(x^*) &= 0\end{aligned}$$

Aufgabe 3 (1)

J. BRACKEN und G. P. McCORMICK, Selected Applications of Nonlinear Programming, John Wiley & Sons, Inc., New York, 1968.

Anzahl der Variablen: 2

Anzahl der Nebenbedingungen: 1 lineare Gleichung
1 nichtlineare Ungleichung

Zu minimierende Zielfunktion:

$$f(x) = (x_1 - 2)^2 + (x_2 - 1)^2$$

Nebenbedingungen:

$$\begin{aligned}g_1(x) &= x_1 - 2x_2 + 1 = 0 \\ g_2(x) &= -\frac{x_1^2}{4} - x_2^2 + 1 \geq 0\end{aligned}$$

Nichtzulässiger Startpunkt:

$$\begin{aligned}x^{(0)} &= (2, 2)^T & g_1(x^{(0)}) &= -1 \\ f(x^{(0)}) &= 1 & g_2(x^{(0)}) &= -4\end{aligned}$$

Ergebnisse:

$$x_1^* = 0.8229$$

$$x_2^* = 0.9114$$

$$f(x^*) = 1.3935$$

$$g_2(x^*) = 0$$

Aufgabe 4 (5)

D.A.PAVIANI, Ph.D. Dissertation, The University of Texas, Austin, Tex., 1969.

Anzahl der Variablen: 3

Anzahl der Nebenbedingungen: 1 nichtlineare Gleichung

1 lineare Gleichung

3 Schranken auf der Variablen

Zu minimierende Zielfunktion:

$$f(x) = 1000 - x_1^2 - 2x_2^2 - x_3^2 - x_1x_2 - x_1x_3$$

Nebenbedingungen:

$$g_1(x) = x_1^2 + x_2^2 + x_3^2 - 25 = 0$$

$$g_2(x) = 8x_1 + 14x_2 + 7x_3 - 56 = 0$$

$$g_i(x) = x_i \geq 0 \quad ; \quad i=1,2,3$$

Nichtzulässiger Startpunkt:

$$x^{(0)} = (2, 2, 2)^T$$

$$f(x^{(0)}) = 976$$

$$g_1(x^{(0)}) = -13$$

$$g_2(x^{(0)}) = 2$$

Ergebnisse:

$$x_1^* = 3.512$$

$$x_3^* = 3.552$$

$$x_2^* = 0.217$$

$$f(x^*) = 961.715$$

Aufgabe 5 (3)

G.K.BARNES, M.S.Thesis, The University of Texas, Austin, Texas, 1967.

Anzahl der Variablen: 2

Anzahl der Nebenbedingungen: 3 nichtlineare Ungleichungen
4 Schranken auf den Variablen

Zu maximierende Zielfunktion:

$$\begin{aligned} f(x) = & 75.196 - 3.8112x_1 + 0.12694x_1^2 - 2.0567 \cdot 10^{-3}x_1^3 \\ & + 1.0345 \cdot 10^{-5}x_1^4 - 6.8306x_2 + 0.030234x_1x_2 \\ & - 1.28134 \cdot 10^{-3}x_2x_1^2 + 3.5256 \cdot 10^{-5}x_2x_1^3 \\ & - 2.266 \cdot 10^{-7}x_2x_1^4 + 0.25645x_2^2 \\ & - 3.4604 \cdot 10^{-3}x_2^3 + 1.3514 \cdot 10^{-5}x_2^4 - \frac{28.106}{x_2+1} \\ & - 5.2375 \cdot 10^{-6}x_1^2x_2^2 - 6.3 \cdot 10^{-8}x_1^3x_2^2 \\ & + 7 \cdot 10^{-10}x_1^3x_2^3 + 3.4054 \cdot 10^{-4}x_1x_2^2 \\ & - 1.6638 \cdot 10^{-6}x_1x_2^3 - 2.8673 \cdot \exp(0.0005x_1x_2) \end{aligned}$$

Nebenbedingungen:

$$\begin{aligned} g_1(x) &= x_1x_2 - 700 \geq 0 \\ g_2(x) &= x_2 - 5\left(\frac{x_1}{25}\right)^2 \geq 0 \\ g_3(x) &= (x_2 - 50)^2 - 5(x_1 - 55) \geq 0 \\ 0 &\leq x_1 \leq 75 \\ 0 &\leq x_2 \leq 65 \end{aligned}$$

Nichtzulässiger Startpunkt:

$$\begin{aligned} x^{(0)} &= (90, 10)^T & g_1(x^{(0)}) &= 300 \\ f(x^{(0)}) &= -82.828 & g_2(x^{(0)}) &= -54.8 \\ & & g_3(x^{(0)}) &= 1425 \end{aligned}$$

Ergebnisse:

$$\begin{array}{ll} x_1^* = 13.563 & g_1(x^*) = 0 \\ x_2^* = 51.610 & g_2(x^*) = 50.14 \\ f(x^*) = 8.198 & g_3(x^*) = 209.78 \end{array}$$

Bemerkungen:

Die Maximierungsaufgabe 5 wurde von den getesteten Verfahren als Minimierungsaufgabe gerechnet. Alle getesteten Verfahren erreichten obiges Ergebnis, welches ein lokales Maximum von Aufgabe 5 darstellt. Das globale Maximum der Aufgabe ist

$$\begin{aligned} x^* &= (75, 65)^T \\ f(x^*) &= 58.903 \end{aligned}$$

Aufgabe 6 (11)

Proctor and Gamble Co. (zitiert in COLVILLE, IBM N.Y. Sci.Center Rept. 320-2949, 1968).

Anzahl der Variablen: 5

Anzahl der Nebenbedingungen: 6 nichtlineare Ungleichungen
10 Schranken auf den Variablen

Zu minimierende Zielfunktion:

$$\begin{aligned} f(x) &= 5.3578547x_3^2 + 0.8356891x_1x_5 \\ &\quad + 37.293239x_1 - 40792.141 \end{aligned}$$

Es ist zu bemerken, daß x_2 und x_4 nicht in der Definition der Zielfunktion enthalten sind.

Nebenbedingungen:

$$\begin{aligned} 0 &\leq 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 \leq 92 \\ 90 &\leq 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 \leq 110 \\ 20 &\leq 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 \leq 25 \end{aligned}$$

$$78 \leq x_1 \leq 102$$

$$33 \leq x_2 \leq 45$$

$$27 \leq x_3 \leq 45$$

$$27 \leq x_4 \leq 45$$

$$27 \leq x_5 \leq 45$$

Zulässiger Startpunkt [Aufgabe 6(a)] :

$$\begin{aligned} x^{(0)} &= (78.62, 33.44, 31.07, 44.18, 35.22)^T \\ f(x^{(0)}) &= -30374 \end{aligned}$$

Nichtzulässiger Startpunkt [Aufgabe 6(b)] :

$$\begin{aligned} x^{(0)} &= (78, 33, 27, 27, 27)^T \\ f(x^{(0)}) &= -32217 \end{aligned}$$

Ergebnisse:

$$\begin{aligned} x^* &= (78, 33, 29.995, 45, 36.775)^T \\ f(x^*) &= -30665.5 \end{aligned}$$

Aufgabe 7 (7)

A.R.COLVILLE, A Comparative Study on Nonlinear Programming Codes, IBM N.Y. Sci.Center Rept. 320-2949, 1968.

Anzahl der Variablen: 3

Anzahl der Nebenbedingungen: 14 nichtlineare Ungleichungen
6 Schranken auf den Variablen

Aufgabe 7 ist ein typisches Beispiel von Aufgaben, in denen Funktionen durch eine in sich abgeschlossene Computeroutine beschrieben sind.

Zu maximierende Zielfunktion:

$$f(x) = 0.063 y_2 y_5 - 5.04 x_1 - 3.36 y_3 - 0.035 x_2 - 10 x_3$$

Nebenbedingungen:

$$0 \leq x_1 \leq 2000$$

$$0 \leq x_3 \leq 120$$

$$0 \leq x_2 \leq 16000$$

$$0 \leq y_2 \leq 5000$$

$$0 \leq y_3 \leq 2000$$

$$85 \leq y_4 \leq 93$$

$$90 \leq y_5 \leq 95$$

$$3 \leq y_6 \leq 12$$

$$0.01 \leq y_7 \leq 4$$

$$145 \leq y_8 \leq 162$$

FORTRAN-Beschreibung der Berechnung von y_2 bis y_8 :

```

      Y(2)=1.6*X(1)
10   Y(3)=1.22*Y(2)-X(1)
      Y(6)=(X(2)+Y(3))/X(1)
      Y2CALC=X(1)*(112.+13.167*Y(6)-0.6667*Y(6)**2)/100.
      IF(ABS(Y2CALC-Y(2))-0.001) 30 , 30 , 20
20   Y(2)=Y2CALC
      GOTO 10
30   CONTINUE
      Y(4)=93.
100  Y(5)=86.35+1.098*Y(6)-0.038*Y(6)**2+0.325*(Y(4)-89.)
      Y(8)=-133.+3.*Y(5)
      Y(7)=35.82-0.222*Y(8)
      Y4CALC=98000.*X(3)/(Y(2)*Y(7)+X(3)*1000.)
      IF(ABS(Y4CALC-Y(4))-0.0001) 300 , 300 , 200
200  Y(4)=Y4CALC
      GOTO 100
300  CONTINUE

```

Zulässiger Startpunkt:

$$x^{(0)} = (1745, 12000, 110)^T$$

$$f(x^{(0)}) = 868.6458$$

Ergebnisse:

$$x^* = (1728.35, 16000, 98.1)^T$$

$$f(x^*) = 1162.03$$

Bemerkung:

Die Maximierungsaufgabe 7 wurde von den getesteten Verfahren als Minimierungsaufgabe gerechnet.

Aufgabe 8 (22)

U.S.Steel Co. (zitiert in HOLZMAN, SRCC Rept.113, 1969).

Anzahl der Variablen: 6

Anzahl der Nebenbedingungen: 4 nichtlineare Ungleichungen
12 Schranken auf den Variablen.

Zu minimierende Zielfunktion:

$$f(x) = 4.3x_1 + 31.8x_2 + 63.3x_3 + 15.8x_4 + 68.5x_5 + 4.7x_6$$

Nebenbedingungen:

$$g_1(x) = 17.1x_1 + 38.2x_2 + 204.2x_3 + 212.3x_4 + 623.4x_5 + 1495.5x_6 \\ - 169x_1x_3 - 3580x_3x_5 - 3810x_4x_5 - 18500x_4x_6 - 24300x_5x_6 \geq b_1$$

$$g_2(x) = 17.9x_1 + 36.8x_2 + 113.9x_3 + 169.7x_4 + 337.8x_5 + 1385.2x_6 \\ - 139x_1x_3 - 2450x_4x_5 - 16600x_4x_6 - 17200x_5x_6 \geq b_2$$

$$g_3(x) = -273x_2 - 70x_4 - 819x_5 + 26000x_4x_5 \geq b_3$$

$$g_4(x) = 159.9x_1 - 311x_2 + 587x_4 + 391x_5 + 2198x_6 - 14000x_1x_6 \geq b_4$$

$$0 \leq x_1 \leq 0.31$$

$$0 \leq x_2 \leq 0.046$$

$$0 \leq x_3 \leq 0.068$$

$$0 \leq x_4 \leq 0.042$$

$$0 \leq x_5 \leq 0.028$$

$$0 \leq x_6 \leq 0.0134$$

Für Aufgabe 8-1:

Für Aufgabe 8-2:

$$b_1 = 4.97$$

$$b_1 = 32.97$$

$$b_2 = -1.88$$

$$b_2 = 25.12$$

$$b_3 = -29.08$$

$$b_3 = -29.08$$

$$b_4 = -78.02$$

$$b_4 = -78.02$$

Startpunkt (zulässig für Aufgabe 8-1; nichtzulässig für Aufgabe 8-2):

$$x_i^{(0)} = 0.01 \quad ; \quad i=1, \dots, 6 \\ f(x^{(0)}) = 1.884$$

Aufgabe 8-1: $g_1(x^{(0)}) = 15.9011$ $g_2(x^{(0)}) = 18.8541$ $g_3(x^{(0)}) = 20.0600$ $g_4(x^{(0)}) = 106.869$	Aufgabe 8-2: $g_1(x^{(0)}) = -12.0989$ $g_2(x^{(0)}) = -8.1459$ $g_3(x^{(0)}) = 20.0600$ $g_4(x^{(0)}) = 106.869$
---	--

Ergebnisse:

Aufgabe 8-1	NEMT-A	NEMT-N	EMT-A	MINI-A MINI-N	MINI(BFS)-A
x_1^*	-6×10^{-7}	-3×10^{-7}	-6×10^{-6}	5×10^{-11}	3×10^{-9}
x_2^*	-4×10^{-6}	-2×10^{-6}	-4×10^{-5}	4×10^{-10}	3×10^{-10}
x_3^*	-8×10^{-6}	-5×10^{-6}	-8×10^{-5}	2×10^{-10}	5×10^{-10}
x_4^*	-2×10^{-6}	-1×10^{-6}	-2×10^{-5}	4×10^{-10}	2×10^{-10}
x_5^*	-1×10^{-5}	-4×10^{-6}	-9×10^{-5}	-7×10^{-10}	-5×10^{-10}
x_6^*	0.00333	0.00333	0.00337	0.00332	0.00332
$f(x^*)$	0.01428	0.01494	0.00308	0.01562	0.01562
$g_1(x^*)$	7×10^{-6}	1×10^{-5}	3×10^{-8}	0.00090	0.00090
$g_2(x^*)$	6.48629	6.48482	6.50927	6.48427	6.48427
$g_3(x^*)$	29.0893	29.0843	29.1638	29.0800	29.0800
$g_4(x^*)$	85.3317	85.3283	85.3916	85.3259	85.3259
Aufgabe 8-2					
x_1^*	0.30986	0.30654	-3×10^{-7}	0.26857	
x_2^*	4×10^{-5}	3×10^{-7}	-2×10^{-6}	8×10^{-12}	
x_3^*	7×10^{-5}	0.00172	0.06293	-4×10^{-12}	
x_4^*	4×10^{-7}	1×10^{-6}	-3×10^{-6}	3×10^{-11}	
x_5^*	0.02643	0.02584	-5×10^{-6}	0.02800	
x_6^*	0.01319	0.01340	0.01345	0.01340	
$f(x^*)$	3.20997	3.25974	4.04611	3.13581	
$g_1(x^*)$	0.06167	0.10385	-3×10^{-8}	-5×10^{-10}	
$g_2(x^*)$	1.63275	1.81904	0.68346	1.25395	
$g_3(x^*)$	7.42652	7.91878	29.0852	6.14799	
$g_4(x^*)$	109.666	109.097	107.589	110.982	

Aufgabe 9 (16)

J.D. PEARSON, On Variable Metric Methods of Minimization, Research Analysis Corps. Rept. RAC - TP - 302, McLean, Va., 1968.

Anzahl der Variablen: 9

Anzahl der Nebenbedingungen: 13 nichtlineare Ungleichungen
1 Schranke auf den Variablen

Aufgabe 9 ist die Maximierung der Fläche eines Hexagons mit maximalem Durchmesser 1.

Zu maximierende Zielfunktion:

$$f(x) = 0.5 (x_1 x_4 - x_2 x_3 + x_3 x_9 - x_5 x_9 + x_5 x_8 - x_6 x_7)$$

Nebenbedingungen:

$$g_1(x) = 1 - x_3^2 - x_4^2 \geq 0$$

$$g_2(x) = 1 - x_9^2 \geq 0$$

$$g_3(x) = 1 - x_5^2 - x_6^2 \geq 0$$

$$g_4(x) = 1 - x_1^2 - (x_2 - x_9)^2 \geq 0$$

$$g_5(x) = 1 - (x_1 - x_5)^2 - (x_2 - x_6)^2 \geq 0$$

$$g_6(x) = 1 - (x_1 - x_7)^2 - (x_2 - x_8)^2 \geq 0$$

$$g_7(x) = 1 - (x_3 - x_5)^2 - (x_4 - x_6)^2 \geq 0$$

$$g_8(x) = 1 - (x_3 - x_7)^2 - (x_4 - x_8)^2 \geq 0$$

$$g_9(x) = 1 - x_7^2 - (x_8 - x_9)^2 \geq 0$$

$$g_{10}(x) = x_1 x_4 - x_2 x_3 \geq 0$$

$$g_{11}(x) = x_3 x_9 \geq 0$$

$$g_{12}(x) = -x_5 x_9 \geq 0$$

$$g_{13}(x) = x_5 x_8 - x_6 x_7 \geq 0$$

$$g_{14}(x) = x_9 \geq 0$$

Nichtzulässiger Startpunkt:

$$x_i^{(0)} = 1 \quad ; \quad i=1, \dots, 9$$

$$f(x^{(0)}) = 0$$

Ergebnisse:

	NEMT-A	NEMT-N	EMT-A	MINI-A MINI-N MINI(BFS)-A
x_1^*	-0.00058	0.00134	3×10^{-6}	-5×10^{-8}
x_2^*	-0.31976	-0.32046	-0.32057	-0.42265
x_3^*	0.86574	0.86577	0.86603	0.86603
x_4^*	-0.50050	-0.50044	-0.50000	-0.50000
x_5^*	-0.00058	-0.00051	-2×10^{-6}	2×10^{-9}
x_6^*	-1.00000	-1.00000	-1.00000	-1.00000
x_7^*	0.86574	0.86670	0.86603	0.86603
x_8^*	0.17974	0.18070	0.17943	0.07735
x_9^*	0.68024	0.67954	0.67943	0.57735
$f(x^*)$	0.866025	0.866024	0.866028	0.866025
$g_1(x^*)$	3×10^{-11}	3×10^{-9}	-2×10^{-6}	-2×10^{-7}
$g_2(x^*)$	0.537273	0.538231	0.538378	0.666667
$g_3(x^*)$	-1×10^{-12}	5×10^{-11}	-2×10^{-6}	-6×10^{-10}
$g_4(x^*)$	2×10^{-8}	8×10^{-8}	-2×10^{-6}	-4×10^{-8}
$g_5(x^*)$	0.537273	0.538227	0.538378	0.666667
$g_6(x^*)$	2×10^{-8}	7×10^{-8}	-2×10^{-6}	1×10^{-7}
$g_7(x^*)$	2×10^{-8}	3×10^{-9}	-2×10^{-6}	4×10^{-7}
$g_8(x^*)$	0.537273	0.536045	0.538373	0.666667
$g_9(x^*)$	1×10^{-8}	6×10^{-8}	-2×10^{-6}	-2×10^{-7}
$g_{10}(x^*)$	0.277117	0.276776	0.277624	0.366025
$g_{11}(x^*)$	0.588908	0.588321	0.588402	0.500000
$g_{12}(x^*)$	0.000394	0.000348	1×10^{-6}	-1×10^{-9}
$g_{13}(x^*)$	0.865631	0.866603	0.866029	0.866025

Bemerkung:

Die Maximierungsaufgabe 9 wurde von den getesteten Verfahren als Minimierungsaufgabe gerechnet.

Aufgabe 10 (4)

J.BRACKEN und G.P.McCORMICK, Selected Applications of Nonlinear Programming, John Wiley & Sons, Inc., N.Y., 1968.

Anzahl der Variablen: 10

Anzahl der Nebenbedingungen: 3 lineare Gleichungen
10 Schranken auf den Variablen

Aufgabe 10 ist eine Aufgabe im chemischen Gleichgewicht bei konstanter Temperatur und konstantem Druck. (Siehe Anhang A).

Zu minimierende Zielfunktion:

$$f(x) = \sum_{i=1}^{10} x_i \left(c_i + \ln \frac{x_i}{\sum_{j=1}^{10} x_j} \right)$$

mit	$c_1 = -6.089$	$c_2 = -17.164$	$c_3 = -34.054$
	$c_4 = -5.914$	$c_5 = -24.721$	$c_6 = -14.986$
	$c_7 = -24.100$	$c_8 = -10.708$	$c_9 = -26.662$
	$c_{10} = -22.179$		

Nebenbedingungen:

$$g_1(x) = x_1 + 2x_2 + 2x_3 + x_6 + x_{10} - 2 = 0$$

$$g_2(x) = x_4 + 2x_5 + x_6 + x_7 - 1 = 0$$

$$g_3(x) = x_3 + x_7 + x_8 + 2x_9 + x_{10} - 1 = 0$$

$$g_{i+3}(x) = x_i \geq 0 \quad ; \quad i=1, \dots, 10$$

Nichtzulässiger Startpunkt:

$$x_i^{(0)} = 0.1 \quad ; \quad i=1, \dots, 10$$

$$f(x^{(0)}) = -20.9603$$

Ergebnisse:

$$\begin{aligned} x_1^* &= 0.04067 & x_6^* &= 0.00069 \\ x_2^* &= 0.14773 & x_7^* &= 0.02740 \\ x_3^* &= 0.78315 & x_8^* &= 0.01795 \\ x_4^* &= 0.00141 & x_9^* &= 0.03731 \\ x_5^* &= 0.48525 & x_{10}^* &= 0.09687 \\ f(x^*) &= -47.7611 \end{aligned}$$

Bemerkung:

Da die Zielfunktion $f(x)$ für negative x_i ($i=1, \dots, 10$) nicht definiert ist, wurde bei NEMT-A, NEMT-N und EMT-A die folgende Funktion als Zielfunktion minimiert:

$$\hat{f}(x) := \sum_{i=1}^{10} x_i \left(c_i + \ln \left| \frac{x_i}{\sum_{j=1}^{10} x_j} \right| \right)$$

$\hat{f}(x)$ stimmt mit $f(x)$ im zulässigen Bereich überein. Die optimalen zulässigen Lösungen sind folglich identisch.

Aufgabe 11 (4a)

Anzahl der Variablen: 10

Anzahl der Nebenbedingungen: 3 nichtlineare Gleichungen.

Aufgabe 11 ist äquivalent zu Aufgabe 10. Die Variablen in Aufgabe 11 entsprechen dabei jeweils dem natürlichen Logarithmus der Variablen in Aufgabe 10.

Zu minimierende Zielfunktion:

$$f(x) = \sum_{i=1}^{10} \left\{ e^{x_i} \left[c_i + x_i - \ln \left(\sum_{j=1}^{10} e^{x_j} \right) \right] \right\}$$

Nebenbedingungen:

$$g_1(x) = e^{x_1} + 2e^{x_2} + 2e^{x_3} + e^{x_6} + e^{x_{10}} - 2 = 0$$

$$g_2(x) = e^{x_4} + 2e^{x_5} + e^{x_6} + e^{x_7} - 1 = 0$$

$$g_3(x) = e^{x_3} + e^{x_7} + e^{x_8} + 2e^{x_9} + e^{x_{10}} - 1 = 0$$

Nichtzulässiger Startpunkt:

$$\begin{aligned}x_i^{(0)} &= -2.3 \quad ; \quad i=1, \dots, 10 \\f(x^{(0)}) &= -21.0145 \\g_1(x^{(0)}) &= -1.2982 \\g_2(x^{(0)}) &= -0.4987 \\g_3(x^{(0)}) &= -0.3984\end{aligned}$$

Ergebnisse:

	NEMT-A	NEMT-N	EMT-A	MINI-A	MINI-N
x_1^*	-3.20231	-3.20231	-3.20230	-3.20231	-3.20233
x_2^*	-1.91236	-1.91236	-1.91230	-1.91237	-1.91237
x_3^*	-0.24443	-0.24443	-0.24444	-0.24443	-0.24443
x_4^*	-6.56117	-6.56118	-6.56105	-6.56118	-6.56124
x_5^*	-0.72310	-0.72310	-0.72310	-0.72310	-0.72310
x_6^*	-7.27426	-7.27429	-7.27432	-7.27423	-7.27493
x_7^*	-3.59721	-3.59721	-3.59714	-3.59724	-3.59722
x_8^*	-4.02027	-4.02027	-4.02015	-4.02032	-4.02033
x_9^*	-3.28836	-3.28836	-3.28828	-3.28838	-3.28839
x_{10}^*	-2.33441	-2.33441	-2.33434	-2.33437	-2.33437
$f(x^*)$	-47.7611	-47.7611	-47.7613	-47.7611	-47.7611
$g_1(x^*)$	2×10^{-16}	5×10^{-13}	4×10^{-6}	2×10^{-7}	5×10^{-8}
$g_2(x^*)$	2×10^{-15}	7×10^{-13}	5×10^{-6}	2×10^{-8}	4×10^{-9}
$g_3(x^*)$	5×10^{-15}	8×10^{-13}	6×10^{-6}	1×10^{-8}	3×10^{-8}

Aufgabe 12 (19)

J.M.GAUTHIER, IBM France (zitiert in COLVILLE, IBM N.Y.Sci. Center Rept. 320-2949, 1968).

Anzahl der Variablen: 16

Anzahl der Nebenbedingungen: 8 lineare Gleichungen
32 Schranken auf den Variablen

Zu maximierende Zielfunktion:

$$f(x) = - \sum_{i=1}^{16} \sum_{j=1}^{16} a_{ij} (x_i^2 + x_i + 1) (x_j^2 + x_j + 1)$$

Nebenbedingungen:

$$\sum_{j=1}^{16} b_{ij} x_j = c_i \quad ; \quad i=1, \dots, 8$$

$$0 \leq x_j \leq 5 \quad ; \quad j=1, \dots, 16$$

Die Werte für a_{ij} , b_{ij} und c_i entnehme man den begleitenden Tafeln.

Nichtzulässiger Startpunkt [Aufgabe 12(a)] :

$$x_j^{(0)} = 10 \quad ; \quad j=1, \dots, 16$$

$$f(x^{(0)}) = -566766$$

Nichtzulässiger Startpunkt [Aufgabe 12(b)] :

$$x_j^{(0)} = 0 \quad ; \quad j=1, \dots, 16$$

$$f(x^{(0)}) = -46$$

Ergebnisse:

$x_1^* = 0.0398$	$x_2^* = 0.7920$	$x_3^* = 0.2029$
$x_4^* = 0.8444$	$x_5^* = 1.2699$	$x_6^* = 0.9347$
$x_7^* = 1.6820$	$x_8^* = 0.1553$	$x_9^* = 1.5679$
$x_{10}^* = 0.0000$	$x_{11}^* = 0.0000$	$x_{12}^* = 0.0000$
$x_{13}^* = 0.6602$	$x_{14}^* = 0.0000$	$x_{15}^* = 0.6743$
$x_{16}^* = 0.0000$		
$f(x^*) = -244.900$		

Bemerkung:

Die Maximierungsaufgabe 12 wurde von den getesteten Verfahren als Minimierungsaufgabe gerechnet.

Data für Aufgabe 12:

j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
b_{1j}	0.22	0.20	0.19	0.25	0.15	0.11	0.12	0.13	1.00							
b_{2j}	-1.46		-1.30	1.82	-1.15		0.80			1.00						
b_{3j}	1.29	-0.89			-1.16	-0.96		-0.49			1.00					
b_{4j}	-1.10	-1.06	0.95	-0.54		-1.78	-0.41					1.00				
b_{5j}				-1.43	1.51	0.59	-0.33	-0.43					1.00			
b_{6j}		-1.72	-0.33		1.62	1.24	0.21	-0.26						1.00		
b_{7j}	1.12			0.31			1.12		-0.36						1.00	
b_{8j}		0.45	0.26	-1.10	0.58		-1.03	0.10								1.00
c_j	2.5	1.1	-3.1	-3.5	1.3	2.1	2.3	-1.5								

ANHANG C

FORTRAN-Subroutine NEMT

Anleitung zur Benutzung von NEMT

SUBROUTINE NEMT (X,GTYPE,ANALYT,N,M,TWO,RAXMIN,AAXMIN,
CONVER,COC,IMAX,NORMF,NORMG,EPS,R)

Bedeutung der Parameter :

- X - Doppeltgenaues Variablenfeld X(N); beim Aufruf von NEMT muß X den Startpunkt enthalten; bei der Rückkehr enthält X die optimale zulässige Lösung.
- GTYPE - Festkommazahlenfeld GTYPE(M); GTYPE(I) muß den Wert 1, 2 oder 3 haben; ist GTYPE(I) = 1 so wird GX(I) als Zielfunktion genommen; ist GTYPE(I) = 2 so wird die Bedingung $GX(I) \geq 0$ gesetzt; ist GTYPE(I) = 3 so wird die Bedingung $GX(I) = 0$ gesetzt.
- ANALYT - Logischer Ausdruck; ist ANALYT = .TRUE. so wird angenommen, daß erste Ableitungen vom Benutzer vorgegeben sind; ist ANALYT = .FALSE. so berechnet NEMT die Ableitungen durch Differenzenbildung.
- N - Festkommazahl; Dimension des Feldes X; $1 \leq N \leq 50$.
- M - Festkommazahl; Dimension der Felder GTYPE und GX; $1 \leq M \leq 50$.
- TWO - Logischer Ausdruck; ist TWO = .TRUE. so werden zwei Funktionswerte pro Ableitung für die numerische Differentiation verwendet; anderenfalls werden vier Funktionswerte genommen.
- RAXMIN - Doppeltgenaue Zahl; relative Genauigkeit der zu liefernden Lösung; vorgeschlagener Wert: 0.000 01 .
- AAXMIN - Doppeltgenaue Zahl; absolute Genauigkeit der zu liefernden Lösung; vorgeschlagener Wert: 0.000 01 .
- CONVER - Logischer Ausdruck; gibt an, ob bei der Rückkehr die Konvergenzkriterien erfüllt sind. Bei Abbruch durch z.B. Überschreiten der maximal erlaubten Zahl von Iterationen ist CONVER = .FALSE.

- COC - Festkommazahl; durch Einstellen von COC wird die Art des Outputs bestimmt; ist COC = 0 so wird kein Output gegeben; ist COC = 1 so wird kurzer Output in jeder Iteration erzeugt und X-Output und G-Output in der ersten und letzten Iteration eines unbedingten Minimierungszyklus; ist COC = 2 so wird zusätzlich X-Output in jeder Iteration erzeugt; eine Erklärung des Outputs ist weiter unten gegeben.
- IMAX - Festkommazahl; maximale vom Benutzer erlaubte Anzahl von Iterationen; ist IMAX = 0 so besteht keine Beschränkung.
- NORMF - Logischer Ausdruck; gibt an ob die Zielfunktion normiert werden soll.
- NORMG - Logischer Ausdruck; gibt an ob die Nebenbedingungen normiert werden sollen.
- EPS - Doppeltgenaue Zahl; EPS ist eine Schranke für die Normierung der Nebenbedingungen, um Division durch sehr kleine Beträge zu vermeiden; vorgeschlagener Wert: 0.01 .
- R - Doppeltgenaue Zahl; Parameter für parametrische äußere Penaltyfunktion im ersten unbedingten Minimierungszyklus (im Null-Zyklus); vorgeschlagener Wert: 0.05 ; bei Versagen des Verfahrens wird vorgeschlagen, die Rechnungen mit kleinerem R zu wiederholen.

Beim Aufruf von NEMT muß der Benutzer eine FORTRAN-Subroutine

FUNCTS (X,GX,N,M)

liefern, in der die Aufgabenfunktionen definiert sind. Ist ANALYT = .TRUE. angegeben, so muß der Benutzer außerdem eine FORTRAN-Subroutine

DERIVE (X,DGDX,N)

liefern, in der die Ableitungen bestimmt sind.

Bedeutung der Parameter von FUNCTS und DERIVE:

X, N, M- Wie oben.

GX - Doppeltgenaues Variablenfeld GX(N); die GX(I) sind die Werte der Aufgabenfunktionen.

DGDX - Doppeltgenaues Variablenfeld DGDX(50,50); DGDX(I,J) ist der Wert der Ableitung der Funktion GX(I) nach X(J) .

Erklärung des Outputs:

Kurzer Output ist eine einzeilige Zusammenfassung der gegenwärtigen Iteration. Es wird die Iterationsnummer, der Wert der Penaltyfunktion und der Betrag des Gradienten im gegenwärtigen Iterationspunkt sowie der Abstand zwischen dem gegenwärtigen und dem vorhergehenden Iterationspunkt ausgegeben.

X - Output ist die Ausgabe des gegenwärtigen Iterationspunktes sowie des Gradienten der Penaltyfunktion in diesem Punkt ausgewertet. Für `ANALYT = .TRUE.` wird der Gradient aus den analytisch vorgegebenen Ableitungen berechnet; andernfalls wird der Gradient durch Differenzenbildung approximiert. In Iteration 0 wird diese Approximation auch für `ANALYT = .TRUE.` berechnet, so daß die analytisch vorgegebenen Ableitungen geprüft werden können.

G - Output ist die Ausgabe der Werte der Aufgabenfunktionen im gegenwärtigen Iterationspunkt.

Folgend ist eine Liste der Subroutine NEMT. Bei der Codierung von NEMT wurden Teile der Subroutine MINI von LOOTSMA [20] übernommen. Die Liste enthält auch das aufrufende Programm und die Subroutinen FUNCTS und DERIVE für die Testaufgabe 3.

```

0010 DOUBLE PRECISION X(2) -
0020 INTEGER GTYPE(3)
0030 LOGICAL CONVER
0040 X(1)=2
0050 X(2)=2
0060 GTYPE(1)=1
0070 GTYPE(2)=3
0080 GTYPE(3)=2
0090 WRITE(6,1)
0100 1 FORMAT(1H)
0110 CALL NENT(X,GTYPE,.TRUE.,2,3,.TRUE.,1,D-5,1,D-5,
0120 * CONVER,1,0,.TRUE.,.TRUE.,1,D-2,5,D-2)
0130 WRITE(6,2) CONVER
0140 2 FORMAT(//8H CONVER=,L1)
0150 END

```

```

0160 SUBROUTINE FUNCTS(X,GX,N,M)
0170 DOUBLE PRECISION X(N),GX(M)
0180 GX(1)=(X(1)-2.D0)**2+X(2)-1.D0**2
0190 GX(2)=X(1)-2.D0*X(2)+1.D0
0200 GX(3)=1.D0-(X(1)**2)/4.D0-X(2)**2
0210 RETURN
0220 END

```

```

0.130 SUBROUTINE DERIVE(X,DGDX,N)
0.140 DOUBLE PRECISION X(N),DGDX(50,50)
0.150 DGDX(1,1)=2 D0*X(1)+4 D0
0.160 DGDX(1,2)=2 D0*X(2)+2 D0
0.170 DGDX(2,1)=1 D0
0.180 DGDX(2,2)=-2 D0
0.190 DGDX(3,1)=-8 D0*X(1)
0.200 DGDX(3,2)=-2 D0*X(2)
0.210 RETURN
0.220 END

```

```

0130      SUBROUTINE NEMT(X, GTYPE, ANALYT, N, M, TWO, RAXMIN,
0140      *  AAXMIN, CONVER, COC, IMAX, NORMF, NORMG, EPS, R)
0150C----MAIN SUBROUTINE INCORPORATES THE NORMED EXTERIOR MOVING TRUNCATI
0160C----PENALTY-FUNCTION METHOD
0170C----MEANING OF PARAMETERS:
0180C----X(N) [DOUBLE PRECISION] - ON CALLING NEMT X(N) MUST CONTAIN THE
0190C----STARTING POINT. ON RETURN X(N) CONTAINS THE CONSTRAINED MINIMUM
0200C----GTYPE(M) [INTEGER] - INDICATES THE NATURE OF THE PROBLEM FUNCTION
0210C----GTYPE=1 INDICATES OBJECTIVE FUNCTION; GTYPE=2 INDICATES
0220C----INEQUALITY CONSTRAINTS. GTYPE=3 INDICATES EQUALITY CONSTRAINT
0230C----ANALYT [LOGICAL] - INDICATES WETHER USER SUPPLIED DERIVATIVES
0240C----ARE AVAILABLE OR NOT
0250C----N [INTEGER] - NUMBER OF VARIABLES
0260C----M [INTEGER] - NUMBER OF PROBLEM FUNCTIONS
0270C----TWO [LOGICAL] - IF TWO IS TRUE, NUMERICAL DIFFERENTIATION TAKES
0280C----TWO FUNCTION VALUES PER DERIVATIVE, OTHERWISE FOUR FUNCTION
0290C----VALUES ARE TAKEN
0300C----RAXMIN [DOUBLE PRECISION] - RELATIVE ACCURACY OF SOLUTION TO BE
0310C----PRODUCED. SUGGESTED VALUE: 1.D-5
0320C----AAXMIN [DOUBLE PRECISION] - ABSOLUTE ACCURACY OF SOLUTION TO BE
0330C----PRODUCED. SUGGESTED VALUE: 1.D-5
0340C----CONVER [LOGICAL] - INDICATES WETHER CONVERGENCE HAS BEEN
0350C----ACHIEVED ON RETURN
0360C----COC [INTEGER] - CONDITIONAL OUTPUT PARAMETER. IF COC=0 NO OUTPUT
0370C----IS GIVEN; IF COC=1 SHORT OUTPUT IS PRODUCED AT EVERY ITERATION
0380C----AND X-OUTPUT AND G-OUTPUT IN THE LAST ITERATION OF AN
0390C----UNCONSTRAINED MINIMIZATION CYCLE. IF COC=2 X-OUTPUT IS GIVEN IN
0400C----ADDITION IN EVERY ITERATION
0410C----IMAX [INTEGER] - MAXIMAL NUMBER OF ITERATIONS PERMITTED BY USER;
0420C----IF IMAX=0 NO RESTRICTION IS IMPOSED
0430C----NORMF [LOGICAL] - INDICATES WETHER OBJECTIVE FUNCTION SHOULD BE
0440C----NORMED
0450C----NORMG [LOGICAL] - INDICATES WETHER CONSTRAINTS SHOULD BE NORMED
0460C----EPS [DOUBLE PRECISION] - LIMIT FOR NORMING OF CONSTRAINTS TO AVOID
0470C----DIVISION BY SMALL NUMBER. SUGGESTED VALUE: 1.D-2
0480C----R [DOUBLE PRECISION] - PARAMETER FOR PARAMETRIC PENALTY FUNCTION
0490C----USED IN FIRST UNCONSTRAINED MINIMIZATION CYCLE (CYCLE=0)
0500C----SUGGESTED VALUE: 5.D-2
0510C----WHEN CALLING NEMT, USER HAS TO SUPPLY SUBROUTINE FUNCTS(X, GX, N, M)
0520C----IN WHICH THE PROBLEM FUNCTIONS GX(M) ARE DEFINED. IF ANALYT=TRUE
0530C----USER HAS TO SUPPLY IN ADDITION A SUBROUTINE DERIVE(X, DGDG, N) IN
0540C----WHICH THE ANALYTICAL DERIVATIVES DGDG(M, N) ARE INTRODUCED INTO THE
0550C----PROGRAM. EXAMPLES FOR FUNCTS AND DERIVE ARE GIVEN.
0560      COMMON /AAA/T, RR, CYCLE, HEAD, /BBB/ANA, T2, RAC, AAC, /CCC/WEPS, NF, NG,
0570      *  , CONSTR, /DDD/CC, IMX, IEFF, /EEE/NC1, NC2, NC3, /FFF/NSEARC
0580      *  /GGG/NEWCYC
0590      DOUBLE PRECISION X(N), R, GX(50), WEPS, EPS, RAXMIN, AAXMIN, RR, T,
0600      *  GT(50), EX(50, 3), TR(3), PEN1, PEN2, PENALT, FX, RAC, AAC
0610      INTEGER GTYPE(M), COC, CYCLE, CC
0620      LOGICAL ANALYT, TWO, CONVER, HEAD, NEWCYC, EXPO, NORMF, NF, NORMG, NG
0630      *  , ANA, T2, CONSTR
0640C
0650      CONVER=.FALSE
0660      IF(N.LE.50.AND.M.LE.50) GOTO 200
0670      WRITE(6,1)
0680 1      FORMAT(' ***ERROR M AND/OR N GREATER THAN 50 ')
0690      GOTO 500
0700 200  RR=R
0710      ANA=ANALYT
0720      T2=TWO
0730      RAC=RAXMIN
0740      AAC=AAXMIN
0750      CC=COC
0760      IMX=IMAX

```

```

0070      NF=NORMF
0080      NG=NORMG
0090      WEPS=EPS
1000      CYCLE=0
1010      IEFF=0
1020      NC1=0
1030      NC2=0
1040      NC3=0
1050      NSEARC=0
1060      HEAD=. TRUE.
1070      NEWCYC=. TRUE.
1080      CONSTR=. FALSE.
1090      DO 210 L=1,M
1100      IF(GTYPE(L).EQ.1) GOTO 210
1110      CONSTR=. TRUE.
1120      GOTO 220
1130 210  CONTINUE
1140 220  DO 230 L=1,M
1150 230  GT(L)=0.D0
1160      PEN1=PENALT(X,0.D0,X,GT,N,M,GTYPE)
1170C
1180C----SEQUENTIAL UNCONSTRAINED MINIMIZATION
1190      DO 400 ICYC=1,50
1200      CALL MINCYC(X,CONVER,N,M,GTYPE,PEN1,GT)
1210      IF(CONVER) GOTO 250
1220      GOTO 500
1230 250  IF(COC.NE.0) CALL OUTPU2(N)
1240      IF(.NOT.CONSTR) GOTO 500
1250C----STORE X(N) AND TRUNCATION LEVEL
1260      IF(CYCLE-1) 300,310,320
1270 300  DO 305 J=1,N
1280 305  EX(J,2)=X(J)
1290      TR(2)=FX-(R/2.D0)
1300      GOTO 330
1310 310  DO 315 J=1,N
1320 315  EX(J,3)=X(J)
1330      TR(3)=T
1340      GOTO 330
1350 320  DO 325 J=1,N
1360      EX(J,1)=EX(J,2)
1370      EX(J,2)=EX(J,3)
1380 325  EX(J,3)=X(J)
1390      TR(1)=TR(2)
1400      TR(2)=TR(3)
1410      TR(3)=T
1420C----EXTRAPOLATE FOR MINIMUM OF NEXT TRUNCATION LEVEL CYCLE
1430 330  NEWCYC=. TRUE.
1440      CYCLE=ICYC
1450      PEN1=PENALT(X,0.D0,X,GT,N,M,GTYPE)
1460      IF(CYCLE.GE.2) GOTO 333
1470      CONVER=. FALSE.
1480      GOTO 400
1490 333  CALL EXTPOL(X,N,EXPO,EX,TR)
1500      IF(.NOT.EXPO) GOTO 380
1510C----TEST EXTRAPOLATED POINT
1520      PEN2=PENALT(X,0.D0,X,GX,N,M,GTYPE)
1530      IF(PEN2.GT.PEN1) GOTO 360
1540      DO 335 J=1,M
1550 335  GT(J)=GX(J)
1560      PEN1=PEN2
1570      IF(COC.NE.0) CALL OUTPU3(N,M,X,GT)
1580C----TEST FOR CONVERGENCE
1590      DO 355 J=1,N
1600      IF(DABS(X(J)-EX(J,3)) LE. RAXMIN*DABS(X(J))+RAXMIN) GOTO 355

```

```

1610      CONVER=.FALSE.
1620      GOTO 400
1630 355  CONTINUE
1640      GOTO 500
1650 360  DO 365 J=1,N
1660 365  X(J)=EX(J,3)
1670 380  DO 385 J=1,N
1680      IF(DABS(X(J)-EX(J,2)).LE.RAXMIN*DABS(X(J))+AAXMIN) GOTO 385
1690      CONVER=.FALSE.
1700      GOTO 400
1710 385  CONTINUE
1720      GOTO 500
1730 400  CONTINUE
1740C
1750 500  RETURN
1760      END

```

```

1770      SUBROUTINE MINCYC(XR,CONVER,N,M,GTYP,PRXR,GXR)
1780C----PERFORMS UNCONSTRAINED FUNCTION MINIMIZATION USING THE BROYDEN
1790C----FORMULA FOR UPDATING THE DIRECTION MATRIX.
1800C----ON CALLING MINCYC, XR(N) MUST CONTAIN THE STARTING POINT AND
1810C----PRXR AND GXR(M) CONTAIN THE VALUES OF THE PENALTY FUNCTION AND
1820C----THE PROBLEM FUNCTIONS RESPECTIVELY.
1830C----ON RETURN XR(N) CONTAINS THE CYCLE MINIMUM AND CONVER INDICATES
1840C----WHETHER CONVERGENCE TO CYCLE MINIMUM HAS BEEN ACHIEVED
1850      COMMON /AAA/T,R,CYCLE,HEAD /BBB/ANALYT,TWO,RAXR,AAXR
1860      * /DDD/COC,IMAX,IEFF
1870      DOUBLE PRECISION XRN,R,GRAD(50),DGRAD(50),DIR(50),SIGMA(50),
1880      * YVEC(50),H(50,50),HY(50),GXR(M),T,RAXR,AAXR,SI,GRADL,PRXR,
1890      * SIGMAY,YHY,DIST,VECVEC,MATVEC,PENALT,YHYSI,HYI,YSS
1900      INTEGER GTYPE(M),COC,COUNT,CYCLE
1910      LOGICAL CONVER,ANALYT,TWO,HEAD
1920C
1930      ITERCT=0
1940      COUNT=0
1950      DIST=0.D0
1960      CONVER=.FALSE.
1970      DO 10 J=1,N
1980      GRAD(J)=0.D0
1990      DIR(J)=0.D0
2000      SIGMA(J)=0.D0
2010 10  YVEC(J)=0.D0
2020      CALL GRADNT(XR,GXR,GRAD,ANALYT,N,M,TWO,GTYP)
2030      IF(ANALYT.AND.IEFF.EQ.0) CALL GRADNT(XR,GXR,DGRAD,.FALSE.,N,M,
2040      * TWO,GTYP)
2050      GRADL=DSQRT(VECVEC(GRAD,GRAD,N))
2060C
2070C----UNCONSTRAINED MINIMIZATION CYCLE
2080      LIM=COUNT+N+1
2090      II=0
2100 50  II=II+1
2110      ITERCT=II-1
2120      SIGMAY=VECVEC(SIGMA,YVEC,N)
2130      IF(ITERCT.NE.0.AND.DABS(SIGMAY).LT.1.D-20) GOTO 150
2140      IF(ITERCT.NE.0) GOTO 30
2150C----INITIALIZE DIRECTION MATRIX
2160      DO 70 I=1,N
2170      DO 60 K=1,N
2180      H(I,K)=0.D0
2190 60  H(K,I)=0.D0
2200 70  H(I,I)=1.D0
2210      GOTO 110
2220 80  DO 90 I=1,N
2230 90  HY(I)=MATVEC(I,H,YVEC,N)
2240      YHY=VECVEC(YVEC,HY,N)
2250      IF(DABS(YHY).LT.1.D-20) GOTO 150
2260      YHYSI=1+YHY/SIGMAY
2270      DO 100 I=1,N
2280      SI=SIGMA(I)/SIGMAY
2290      HYI=HY(I)/SIGMAY
2300      YSS=YHYSI*SI
2310C----UPDATE DIRECTION MATRIX
2320      DO 100 K=1,I
2330      H(I,K)=H(I,K)-HY(K)*SI-HYI*SIGMA(K)+YSS*SIGMA(K)
2340C      H(I,K)=H(I,K)+SI*SIGMA(K)-HY(I)*HY(K)/YHY
2350 100 H(K,I)=H(I,K)
2360 110 DO 115 J=1,N
2370 115 DIR(J)=MATVEC(J,H,GRAD,N)
2380      IF(VECVEC(GRAD,DIR,N).LE.0.D0) GOTO 130
2390      DO 120 J=1,N
2400 120 DIR(J)=-DIR(J)

```

```

2410 130 IF(COC.NE.0) CALL OUTPU1(ITERCT,CONVER,PRXR,GRADL,DIST,XR,GRAD,
2420 * DGRAD,GXR,N,M,GTYPE)
2430 IEFF=IEFF+1
2440 IF(IEFF.EQ.IMAX) RETURN
2450C----MINIMIZATION ALONG LINE
2460 CALL LINMIN(XR,PRXR,GXR,GRAD,DIR,SIGMA,YVEC,M,N,GTYPE)
2470 GRADL=DSQRT(VECVET(GRAD,GRAD,N))
2480 DIST=DSQRT(VECVET(SIGMA,SIGMA,N))
2490C---TEST THE ACCURACY
2500 IF(DIST.LE.1.D-10.OR.GRADL.LE.1.D-10) GOTO 150
2510 DO 135 J=1,M
2520 IF(DABS(SIGMA(J)).LE.RAXR*DABS(XR(J))+AAXR)GOTO 135
2530 COUNT=ITERCT
2540 LIM=COUNT+N+1
2550 GOTO 140
2560 135 CONTINUE
2570 140 IF(II.LT.LIM) GOTO 50
2580C
2590 150 CONVER=.TRUE.
2600 IF(COC.NE.0) CALL OUTPU1(ITERCT,CONVER,PRXR,GRADL,DIST,XR,GRAD,
2610 * DGRAD,GXR,N,M,GTYPE)
2620 RETURN
2630 END

```

```

2640      DOUBLE PRECISION FUNCTION PENALT(P,T,Q,GT,N,M,GTYP)
2650C----COMPUTES THE PROBLEM FUNCTIONS GT(M) AND THE PENALTY FUNCTION AT
2660C----THE POINT P(N)+T*Q(N). FOR CYCLE=0 THE PARAMETRIC PENALTY
2670C----FUNCTION, OTHERWISE THE (NORMED) EXTERIOR MOVING TRUNCATION
2680C----PENALTY FUNCTION IS COMPUTED
2690      COMMON /AAA/TRUNC,R,CYCLE,HEAD /CCC/WEPS,NORMF,NORMG,PENAL1,CN
2700      * /EEE/NC1,NC2,NC3 /GGG/NEWCYC /HHH/NORM(50)
2710      DOUBLE PRECISION P(N),Q(N),GT(M),XT(50),T,R,TEX,NORM,WEPS,
2720      * PENAL1,TRUNC,ALTPEN,MF
2730      INTEGER GTYPE(M),CYCLE
2740      LOGICAL NEWCYC,NORMF,NORMG,HEAD,CN
2750C
2760      PENAL1=0.D0
2770      TEX=0.D0
2780      DO 10 J=1,N
2790 10  XT(J)=P(J)+T*Q(J)
2800      IF(CYCLE.NE.0.OR..NOT NEWCYC) GOTO 30
2810      DO 20 I=1,M
2820 20  NORM(I)=1.D0
2830C
2840C----COMPUTE PROBLEM FUNCTIONS
2850 30  CALL FUNCTS(XT,GT,N,M)
2860      NC2=NC2+1
2870C
2880C----COMPUTE PENALTY FUNCTION
2890      DO 150 I=1,M
2900      IND=GTYP(I)+1
2910      GOTO(150,80,120,130),IND
2920 80  PENAL1=PENAL1+GT(I)
2930      IF(CYCLE.EQ.0) GOTO 150
2940      IF(.NOT.NEWCYC) GOTO 110
2950      IF(CYCLE.EQ.1) GOTO 82
2960C----DETERMINE TRUNCATION LEVEL AT BEGINNING OF NEW CYCLE
2970      IF(NORM(I).LT.1.D-3) GOTO 81
2980      TRUNC=TRUNC+ALTPEN/(NORM(I)**2*(PENAL1-TRUNC)*2.D0)
2990 81  IF(TRUNC.LT.PENAL1) TRUNC=PENAL1
3000      GOTO 90
3010 82  TRUNC=ALTPEN
3020 90  IF(.NOT.NORMF) GOTO 110
3030C----DETERMINE OBJECTIVE FUNCTION NORM AT BEGINNING OF NEW CYCLE
3040      MF=0.D0
3050      DO 85 J=1,M
3060      IF(GTYP(J)-2) 85,83,84
3070 83  IF(GT(J).GE.0.D0) GOTO 85
3080 84  MF=MF+1.D0
3090 85  CONTINUE
3100      IF(MF.LT.1.D0) MF=1.D0
3110      IF(DABS(PENAL1).LT.1.D4) GOTO 86
3120      NORM(I)=MF/1.D4
3130      GOTO 110
3140 86  IF(DABS(PENAL1).GT.MF) NORM(I)=MF/DABS(PENAL1)
3150 110  PENALT=0.D0
3160      IF(TRUNC-PENAL1.LT.0.D0) PENALT=(NORM(I)*(TRUNC-PENAL1))**2
3170      GOTO 150
3180 120  IF(.NOT.NEWCYC.OR.CYCLE.EQ.0.OR..NOT.NORMG) GOTO 125
3190C----DETERMINE CONSTRAINTS NORMS AT BEGINNING OF NEW CYCLE
3200C----INEQUALITY CONSTRAINTS
3210      IF(GT(I).GT.-1.D0.AND GT(I).LT.-WEPS) GOTO 121
3220      IF(GT(I).GE.-WEPS) NORM(I)=1.D0/WEPS**2
3230      GOTO 125
3240 121  NORM(I)=1.D0/GT(I)**2
3250 125  IF(GT(I).LT.0.D0) GOTO 140
3260      GOTO 150
3270 130  IF(.NOT.NEWCYC.OR.CYCLE.EQ.0.OR..NOT.NORMG) GOTO 140

```

```

3280C----EQUALITY CONSTRAINTS
3290      IF(DABS(GT(I))>.LT.1.D0.AND.DABS(GT(I))>.GT.WEPS) GOTO 131
3300      IF(DABS(GT(I))>.LE.WEPS) NORM(I)=1.D0/WEPS**2
3310      GOTO 140
3320 131  NORM(I)=1.D0/GT(I)**2
3330 140  TEX=TEX+NORM(I)*GT(I)**2
3340 150  CONTINUE
3350      NEWCYC=.FALSE
3360      IF(CYCLE.NE 0) GOTO 200
3370C----PARAMETRIC PENALTY FUNCTION
3380      PENALT=PENALT1+TEX/R
3390      ALTPEN=PENALT
3400      RETURN
3410C----MOVING TRUNCATION PENALTY FUNCTION
3420 200  PENALT=PENALT1+TEX
3430      ALTPEN=PENALT
3440      RETURN
3450      END

```

```

3460      SUBROUTINE GRADNT(XT,GT,DPTDX,ANALYT,N,M,TWO,GTYPE)
3470C----COMPUTES THE GRADIENT DPTDX(N) OF THE PENALTY FUNCTION AT THE
3480C----POINT XT(N). BEFORE CALLING THE SUBROUTINE GRADNT THE VECTOR GT(M)
3490C----MUST CONTAIN THE VALUES OF THE PROBLEM FUNCTIONS AT THIS POINT
3500      COMMON /AAA/TRUNC,R,CYCLE,HEAD /EEE/NC1,NC2,NC3 /HHH/NORM(50)
3510      DOUBLE PRECISION XT(N),GT(M),DPTDX(N),GDELTA(50),DGDXX(50),NORM,
3520      * DIFF(50,2),DGDXX(50,50),R,XTJ,HJ,DJ,DPJ,TRUNC,FX,DFXJ
3530      INTEGER GTYPE(M),UPPER,CYCLE
3540      LOGICAL ANALYT,TWO,HEAD
3550      UPPER=2
3560      IF(TWO) UPPER=1
3570      IF(.NOT.ANALYT) GOTO 20
3580C
3590C----COMPUTE USERS DERIVATIVES
3600      DO 10 J=1,N
3610      DO 10 I=1,M
3620 10    DGDXX(I,J)=0.D0
3630      CALL DERIVE(XT,DGDXX,N)
3640      NC3=NC3+1
3650C
3660C----STORE DERIVATIVES WITH RESPECT TO XTJ
3670 20    DO 300 J=1,N
3680 30    XTJ=XT(J)
3690      IF(.NOT.ANALYT) GOTO 50
3700      DO 40 I=1,M
3710 40    DGDXX(I)=DGDXX(I,J)
3720      GOTO 100
3730C
3740C----COMPUTE NUMERICAL DERIVATIVES
3750 50    HJ=1.D-3*DABS(XTJ)+1.D-6
3760      DO 70 K=1,UPPER
3770      DJ=HJ/K
3780      NC1=NC1+2
3790      XT(J)=XTJ+DJ
3800      CALL FUNCTS(XT,GDELTA,N,M)
3810      DO 60 I=1,M
3820 60    DIFF(I,K)=GDELTA(I)
3830      XT(J)=XTJ-DJ
3840      CALL FUNCTS(XT,GDELTA,N,M)
3850      DO 70 I=1,M
3860 70    DIFF(I,K)=(DIFF(I,K)-GDELTA(I))/(2.D0*DJ)
3870      XT(J)=XTJ
3880      DO 90 I=1,M
3890 90    IF(TWO) GOTO 80
3900      DGDXX(I)=(4.D0*DIFF(I,2)-DIFF(I,1))/3.D0
3910      GOTO 90
3920 80    DGDXX(I)=DIFF(I,1)
3930 90    CONTINUE
3940C
3950C----COMPUTE GRADIENT OF PENALTY FUNCTION
3960 100    DPJ=0.D0
3970      FX=0.0D0
3980      DFXJ=0.0D0
3990      IF(CYCLE.NE 0) GOTO 160
4000C----PARAMETRIC PENALTY FUNCTION
4010      DO 140 I=1,M
4020      INDEX=GTYPE(I)+1
4030      GOTO (140,110,120,130),INDEX
4040 110    DPJ=DPJ+DGDXX(I)
4050      GOTO 140
4060 120    IF(GT(I).GE.0.D0) GOTO 140
4070 130    DPJ=DPJ+2.D0*DGDXX(I)*GT(I)/R
4080 140    CONTINUE
4090 150    DPTDX(J)=DPJ

```

```

4100      GOTO 300
4110C---MOVING TRUNCATION PENALTY FUNCTION
4120 160  DO 240 I=1,M
4130      INDEX=GTYPX(I)+1
4140      GOTO(240,210,220,230) INDEX
4150 210  FX=FX+GT(I)
4160      DFXJ=DFXJ+IND(I)
4170      DFTDX(J)=2 BB+FX+YFUND*(YFUND+H*F+I)*2
4180      GOTO 240
4190 220  IF(NT(I) GE 0) GOTO 240
4200 230  DPJ=DPJ+2 BB+DCIND(I)*G(I)*XNORM(I)
4210 240  CONTINUE
4220 250  DFTDX(J)=DFTDX(J)+DPJ
4230      IF(FUNC-FX GT 0) GO TO 210
4240C
4250 300  CONTINUE
4260      RETURN
4270      END

```

```

4100 SUBROUTINE LINMIN(PPOINT, GPOINT, GRADPR, S, DIFFP, DIFFG, M, N,
4200 * GTYPE)
4300----SEARCHING A MINIMUM WITH RELATIVE AND ABSOLUTE ACCURACY RACC AND
4310----RACC RESPECTIVELY ALONG THE DIRECTION S(N) EMANATING FROM POINT(N)
4320----PRPNT AND GPOINT(M) MUST CONTAIN THE VALUES OF THE PENALTY FUNCTION
4330----AND THE PROBLEM FUNCTIONS AT POINT(N).
4340----GRADPR(N) IS THE GRADIENT OF THE PENALTY FUNCTION AT POINT(N).
4350----ON RETURN FROM LINMIN THE LINE MINIMUM IS CONTAINED IN POINT(N),
4360----AND PRPNT, GPOINT(M) AND GRADPR(N) CONTAIN THE CORRESPONDING DATA.
4370----THE DIFFERENCE BETWEEN THE LINE MINIMUM AND THE STARTING POINT OF
4380----THE SEARCH IS STORED IN DIFFP(N) THE DIFFERENCE OF THE
4390----CORRESPONDING GRADIENTS IS CONTAINED IN DIFFG(N).
4400 COMMON /BBB/ANALYT, TWO, RACC, AACC /FFF/NSEARC
4410 DOUBLE PRECISION POINT(N), GPOINT(M), GRADPR(N), S(N), DIFFP(N),
4420 * DIFFG(N), GA(50), GB(50), GC(4), GD(50), PRPNT, RACC, AACC, LAMBDA,
4430 * DESCNT, A, B, C, D, PRA, PRB, PRC, PRD, PA, PC, PENALT, VECVEC, TEST
4440 INTEGER GTYPE(M)
4450 LOGICAL ANALYT, TWO
4460
4470 NSEARC=NSEARC+1
4480 DESCNT=VECVEC(GRADPR, S, M)
4490 IF(DESCNT.NE.0.D0) GOTO 5
4500 DO 2 I=1, N
4510 DIFFP(I)=0.D0
4520 2 DIFFG(I)=0.D0
4530 RETURN
4540 5 LAMBDA=DABS(2.D0*PRPNT/DESCNT)
4550 IF(LAMBDA.NE.1.D0) LAMBDA=1.D0
4560 A=0.D0
4570 B=0.D0
4580 C=0.D0
4590 PRA=PRPNT
4600 PRB=PRPNT
4610 PRC=PRPNT
4620 DO 10 I=1, M
4630 GA(I)=GPOINT(I)
4640 GB(I)=GPOINT(I)
4650 10 GC(I)=GPOINT(I)
4660
4670----MOVE IN SEARCH DIRECTION
4680 20 DO 40 NSTEPS=1, 50
4690 C=B+LAMBDA
4700 PRC=PENALT(PPOINT, C, S, GC, N, M, GTYPE)
4710 IF(PRC.GT. PRB) GOTO 50
4720 LAMBDA=2*LAMBDA
4730 CALL SHIFT(A, B, PRA, PRB, GA, GB, M)
4740 CALL SHIFT(B, C, PRB, PRC, GB, GC, M)
4750 40 CONTINUE
4760
4770 50 IF(.NOT.(A.EQ.0.D0.AND.B.EQ.0.D0.AND.C.GT.0.D0.AND.DESCNT.LT.0.D0
4780 * .AND.PR.B.LT.PRC)) GOTO 90
4790
4800----REDUCE SEARCH INTERVAL
4810 DO 80 NSTEPS=1, 50
4820 D=0.5D0*C
4830 PRD=PENALT(PPOINT, D, S, GD, N, M, GTYPE)
4840 IF(PRD.LE. PRB) GOTO 70
4850 CALL SHIFT(C, D, PRC, PRD, GC, GD, M)
4860 GOTO 80
4870 70 CALL SHIFT(B, D, PRB, PRD, GB, GD, M)
4880 GOTO 90
4890 80 CONTINUE
4900
4910 90 IF(.NOT.(A.LT.B.AND.B.LT.C.AND.PRA.GE. PRB.AND. PRB.LE. PRC)) GOTO 200

```

```

45200
49300---QUADRATIC INTERPOLATION ON THREE POINTS A,B,C
4940    DO 180 NSTEPS=1,50
4950    PC= A-B*(PRC-FRB)
4960    IF(PC.NE.0.00) GOTO 140
4970    D=0.500*(B+C)
4980    GOTO 110
4990 100 PA=(B-C)*(PRA-FRB)
5000    D=0.500*(A+B+PC+(C+D)*(PRC-FRB)+PA+PC)
5010 110 PRD=PLNALT(POINT,PR,GRAD,N,TYPE)
5020    IF(D.EQ.0.00 AND PRD.GE.FRB OR DABS(PRD-FRB).GT
5030    * RACC*DABS(PRD)+ALCO) GOTO 140
5040    IF(PRD.LT.FRB) CALL SHIFT(B,D,PRB,PRD,GB,GD,M)
5050    GOTO 200
5060 140 IF(D.GE.B) GOTO 150
5070    IF(PRD.GE.FRB) GOTO 150
5080    CALL SHIFT(C,D,PRC,PRD,GC,GD,M)
5090    CALL SHIFT(B,D,PRB,PRD,GB,GD,M)
5100    GOTO 180
5110 150 CALL SHIFT(C,D,PRA,PRD,GA,GD,M)
5120    GOTO 180
5130 160 IF(PRD.GE.FRB) GOTO 170
5140    CALL SHIFT(A,B,PRA,PRD,GA,GD,M)
5150    CALL SHIFT(B,D,PRB,PRD,GB,GD,M)
5160    GOTO 180
5170 170 CALL SHIFT(C,D,PRC,PRD,GC,GD,M)
5180 180 CONTINUE
51900
52000---MOVE ITERATION POINT
5210 200 PRPNT=FRB
5220    DO 210 J=1,N
5230    DIFFG(J)=B*G(J)
5240    DIFFG(J)=GRADPR(J)
5250 210 POINT(J)=POINT(J)+DIFFG(J)
52600
5270    DO 220 I=1,M
5280 220 GPOINT(I)=GB+I
5290    CALL GRADNT(POINT,GPU,N,GRADER,APPL,Z,N,M,TWO,TYPE)
5300    DO 230 J=1,N
5310 230 DIFFG(J)=GRADPP(J)-DIFFG(J)
5320    RETURN
5330    END

```

```

5150 SUBROUTINE OUTPUT/ITERCT, CONVER, PRXR, GRADL, DIST, XR, GRAD,
5150 * DGRAD, GXR, N, N, GTYPE)
5160C-----GENERATES THE MAIN PRINTED OUTPUT
5170 COMMON /AAAT, R, CYCLE, HEAD, /BBB/ANALYT, TWO, RAC, AAC, /CCC/EPS, NF, NG
5180 * /FF/CONST, /DD/CCC, /THH/IEFF
5190 DOUBLE PRECISION T, PRXR, GRADL, DIST, XR(N), GRAD(N), DGRAD(N), GXR(M), R
5200 EPS, RAC, AAC, FX
5210 INTEGER CYCLE, CCC, GTYPE, M
5220 LOGICAL ANALYT, CONVER, HEAD, TWO, NF, NG, CONST
5230C
5240 IF(ITERCT.NE.0) GOTO 30
5250 IF(CCC.LE.0) GOTO 20
5260 DO 10 J=1, M
5270 10 WRITE(6, 100) J, GTYPE(J)
5280 100 FORMAT(6H GTYPE, 12, 1H=, 11)
5290 IF(CONST) WRITE(6, 110) NF, NG
5300 110 FORMAT(47H NORMF=, L1, 7H NORMG=, L1)
5310 IF(NG.AND.CONST) WRITE(6, 120) EPS
5320 120 FORMAT(15H EPS FOR NORMG=, D11, 3)
5330 WRITE(6, 130) ANALYT
5340 130 FORMAT( 8H ANALYT=, L1)
5350 IF( NOT ANALYT) WRITE 6, 135) TWO
5360 135 FORMAT(1H+, 10X, 4H TWO=, L1)
5370 WRITE(6, 140) RAC, AAC
5380 140 FORMAT( 8H RACMIN=, D11, 3, 8H AACMIN=, D11, 3)
5390 IF( NOT CONST) GOTO 30
5400 WRITE(6, 150) R
5410 150 FORMAT( 24H R FOR PARAMETRIC CYCLE=, D11, 3)
5420 WRITE(6, 160)
5430 160 FORMAT(//26H BEGIN OF PARAMETRIC CYCLE)
5440 GOTO 30
5450 20 WRITE(6, 170) CYCLE, T
5460 170 FORMAT(//15H BEGIN OF CYCLE 13, 15H FOR T EQUAL TO, D19, 6)
5470 30 IF( NOT HEAD) GOTO 40
5480C
5490C-----SHORT OUTPUT
5500 WRITE(6, 180)
5510 180 FORMAT(//62H ITERATION      PENALTY VALUE      GRADIENT LENGTH      D
5520 * DISTANCE)
5530 40 WRITE(6, 190) IEFF, PRXR, GRADL, DIST
5540 190 FORMAT(1X, 16, 3D19, 6)
5550 HEAD=.FALSE.
5560C
5570 IF(IEFF.NE.0.AND. NOT CONVER.AND.CCC.NE.2) RETURN
5580C
5590C-----OUTPUT
5600 WRITE(6, 200)
5610 200 FORMAT(//47H VARIABLE      SOLUTION VECTOR      GRADIENT OF PENF)
5620 IF(ANALYT.AND.IEFF.EQ.0) WRITE(6, 210)
5630 210 FORMAT(1H+, 49X, 19H GRAD DIFFERENCES))
5640 DO 50 J=1, N
5650 WRITE(6, 220) J, XR(J), GRAD(J)
5660 220 FORMAT(4H X 13, 2D19, 6)
5670 IF(ANALYT.AND.IEFF.EQ.0) WRITE(6, 230) DGRAD(J)
5680 230 FORMAT(1H+, 44X, 1D19, 6)
5690 50 CONTINUE
5700 HEAD=.TRUE.
5710C
5720 IF(IEFF.NE.0.AND. NOT CONVER) RETURN
5730C
5740C-----OUTPUT
5750 WRITE(6, 240)
5760 240 FORMAT(//27H FUNCTION      FUNCTION VALUES)
5770 DO 60 I=1, M

```

```

5980      WRITE(6,250) I,GXR(I)
5990 250  FORMAT(5H   GX, I2, D19.6)
6000 60   CONTINUE
6010      HEAD=.TRUE.
6020C
6030      IF(CYCLE.NE.0) GOTO 70
6040      IF(CONVER.AND.CONSTR) WRITE(6,260)
6050 260  FORMAT(/24H END OF PARAMETRIC CYCLE)
6060      RETURN
6070 70   IF(CONVER) WRITE(6,270) CYCLE,T
6080 270  FORMAT(/13H END OF CYCLE, I3, 15H FOR T EQUAL TO, I D19.6)
6090      RETURN
6100      END

```

```

6115      SUBROUTINE OUTPUT3(N,M,XMIN,GMIN)
6120C----GENERATES THE PRINTED OUTPUT AFTER EVERY SUCCESSFULL EXTRAPOLATION
6130      COMMON /AAA/T,R,CYCLE,HEAD
6140      DOUBLE PRECISION XMIN(N),GMIN(M),T,R
6150      INTEGER CYCLE
6160      LOGICAL HEAD
6170      IC=CYCLE-1
6180      WRITE(6,1) IC
6190 1      FORMAT(/21H EXTRAPOLATION,CYCLE ,I2)
6200C
6210C----X-OUTPUT
6220      WRITE(6,2)
6230 2      FORMAT(/27H VARIABLE      SOLUTION VECTOR)
6240      DO 10 J=1,N
6250 10     WRITE(6,3) J,XMIN(J)
6260 3      FORMAT(4H   X,I3,D19.6)
6270C
6280C----G-OUTPUT
6290      WRITE(6,4)
6300 4      FORMAT(/27H FUNCTION      FUNCTION VALUES)
6310      DO 20 I=1,M
6320 20     WRITE(6,5) I,GMIN(I)
6330 5      FORMAT(5H   GX,I2,D19.6)
6340      HEAD= .TRUE.
6350C
6360      RETURN
6370      END
6380      SUBROUTINE OUTPUT2(NV)
6390C----GENERATES ADDITIONAL OUTPUT AFTER EVERY COMPLETED CYCLE
6400      COMMON /EEE/NC1,NC2,NC3 /FFF/NSEARC
6410      WRITE(6,1) NC1
6420 1      FORMAT(/15,39H EVALUATIONS OF *FUNCTS* FOR NUM. DIFF.)
6430      WRITE(6,2) NC2
6440 2      FORMAT(1X,I4,42H EVALUATIONS OF *FUNCTS* FOR LINE SEARCHES)
6450      IF(NC3.EQ.0) GOTO 10
6460      WRITE(6,3) NC3
6470 3      FORMAT(1X,I4,24H EVALUATIONS OF *DERIVE*)
6480 10     I=NC1+NC2+NC3*NV
6490      WRITE(6,4) I
6500 4      FORMAT(1X,I4,32H EQUIVALENT FUNCTION EVALUATIONS)
6510      I=NC2/NSEARC
6520      WRITE(6,5) I
6530 5      FORMAT(1X,I4,41H EVALUATIONS OF *FUNCTS* PER LINE MINIMUM)
6540      RETURN
6550      END

```

```

6560      SUBROUTINE SHIFT(Y,Z,PRY,PRZ,GY,GZ,M)
6570C----SHIFTS VALUES IN LINMIN
6580      DOUBLE PRECISION Y,Z,PRY,PRZ,GY(M),GZ(M)
6590      Y=Z
6600      PRY=PRZ
6610      DO 10 I=1,M
6620 10    GY(I)=GZ(I)
6630      RETURN
6640      END

```

```

650      DOUBLE PRECISION FUNCTION VECVEC(A,B,N)
660C-----MULTIPLIES N-VECTOR BY N-VECTOR
670      DOUBLE PRECISION A(N),B(N),VEC1
680      VEC1=0.D0
690      DO 10 K=1,N
700 10    VEC1=A(K)*B(K)+VEC1
710      VECVEC=VEC1
720      RETURN
730      END

```

```

6740      DOUBLE PRECISION FUNCTION MATVEC(I,A,B,N)
6750C----MULTIPLIES I-TH ROW VECTOR OF (L,N)-MATRIX BY N-VECTOR
6760      DOUBLE PRECISION A(50,50),B(N),MAT1
6770      MAT1=0.D0
6780      DO 10 K=1,N
6790 10    MAT1=A(I,K)*B(K)+MAT1
6800      MATVEC=MAT1
6810      RETURN
6820      END

```

```

6930      SUBROUTINE EXTPOL(X,N,EXPO,EX,TR)
6940C----EXTRAPOLATES TOWARD MINIMUM OF NEXT UNCONSTRAINED CYCLE.
6950C----EX(N,3) AND TR(3) MUST CONTAIN THE MINIMA OF THE LAST THREE CYCLES.
6960C----IF ONLY TWO CYCLES HAVE BEEN COMPLETED, LINEAR EXTRAPOLATION
6970C----IS CARRIED OUT. ON RETURN X(N) CONTAINS THE EXTRAPOLATED POINT.
6980C----EXPO INDICATES WHETHER EXTRAPOLATION HAS BEEN CARRIED THROUGH.
6990      COMMON /AAA/T,R,CYCLE,HEAD
7000      DOUBLE PRECISION X(N),T,TR(3),TR2(3),EX(50,3),MM(4),AA(50,3),R
7010      LOGICAL EXPO,HEAD
7020      INTEGER CYCLE
7030C
7040      EXPO= .FALSE.
7050      IF(CYCLE.GT.2) GOTO 100
7060C
7070C----LINEAR EXTRAPOLATION
7080      MM(4)=TR(3)-TR(2)
7090      IF(DABS(MM(4)).LT.1.D-5) GOTO 300
7100      DO 10 I=1,N
7110      AA(I,1)=(TR(3)*EX(I,2)-TR(2)*EX(I,3))/MM(4)
7120      AA(I,2)=(EX(I,3)-EX(I,2))/MM(4)
7130      X(I)=AA(I,1)+AA(I,2)*T
7140      GOTO 200
7150C
7160C----QUADRATIC EXTRAPOLATION
7170      DO 30 J=1,3
7180      TR2(J)=TR(J)**2
7190      MM(1)=TR(2)*TR2(3)-TR(3)*TR2(2)
7200      MM(2)=TR(1)*TR2(3)-TR(3)*TR2(1)
7210      MM(3)=TR(1)*TR2(2)-TR(2)*TR2(1)
7220      MM(4)=MM(1)-MM(2)+MM(3)
7230      IF(DABS(MM(4)).LT.1.D-5) GOTO 300
7240      DO 40 I=1,N
7250      AA(I,1)=(MM(1)*EX(I,1)-MM(2)*EX(I,2)+MM(3)*EX(I,3))/MM(4)
7260      AA(I,2)=((EX(I,2)*TR2(3)-EX(I,3)*TR2(2))-(EX(I,1)*TR2(3)
7270      * -EX(I,3)*TR2(1))+EX(I,1)*TR2(2)-EX(I,2)*TR2(1))/MM(4)
7280      AA(I,3)=((TR(2)*EX(I,3)-TR(3)*EX(I,2))-(TR(1)*EX(I,3)
7290      * -TR(3)*EX(I,1)+(TR(1)*EX(I,2)-TR(2)*EX(I,1))/MM(4)
7300      X(I)=AA(I,1)+AA(I,2)*T+AA(I,3)*T**2
7310C
7320      EXPO= .TRUE.
7330      RETURN
7340      END

```


LEBENS LAUF

Heinz Roland Weistroffer wurde am 24. Januar 1950 als drittes Kind des Ingenieurs Rudolf P. Weistroffer und dessen Frau Ursula geb. Krause in Herne, Nordrhein-Westfalen, geboren. 1956 zog die Familie nach Khartoum, der Hauptstadt der gerade unabhängig gewordenen Republik Sudan.

Von 1958 bis 1965 besuchte Roland Weistroffer das Comboni College Khartoum, eine italienische Missionsschule mit Unterrichtssprache Englisch. Ostern 1965 ging er nach Plön /Schleswig-Holstein auf das Internatsgymnasium Schloß Plön, wo er im Mai 1968 die Reifeprüfung bestand.

Im Wintersemester 1968/69 studierte er Mathematik an der Ruhruniversität Bochum und vom Sommersemester 1969 bis Sommersemester 1971 an der Freien Universität Berlin, wo er im Oktober 1970 die Vorprüfung zur Diplomprüfung in Mathematik bestand.

Von August 1971 bis Dezember 1972 studierte Roland Weistroffer weiter Mathematik an der Duke University in Durham /North Carolina und erhielt dort den akademischen Grad "Master of Arts".

Nach Berlin zurückgekehrt, nahm er Mitte 1973 an der Freien Universität, unter Anleitung von Herrn Professor Dr. Udo Schendel, die vorliegende Arbeit in Angriff.

1973 arbeitete Roland Weistroffer fünf Monate als Praktikant in der Datenverarbeitung der Firma Deutsche Industrie-Anlagen-Gesellschaft m. b.H. Berlin. Danach bezog er ein Stipendium nach dem Gesetz für Graduiertenförderung.

Seit Juni 1974 ist er mit Linda geb. Duke verheiratet.

